AD-A081 596 MCDONNELL DOUGLAS ASTRONAUTICS CO HUNTINGTON BEACH CA F/6 9/2
AERODYNAMIC COMPUTER CODE FOR COMPUTING PRESSURE LOADS ON A MIS--ETC(U)
DEC 79 K K WANG N00014-75-C-0462
UNCLASSIFIED MDC-G8223 NL

1 of 3
AD
A081596

LEVEL IV

12

MDC G8223

ADA081596

AERODYNAMIC COMPUTER CODE FOR COMPUTING PRESSURE LOADS ON A MISSILE AT HIGH SUPERSONIC SPEEDS AND GENERALIZED ATTITUDES

DTIC
S ELECTE D
MAR 6 1980
C

Prepared by Kenneth K. Wang

MCDONNELL DOUGLAS ASTRONAUTICS COMPANY
5301 BOLSA AVENUE
HUNTINGTON BEACH, CALIFORNIA 92647

NOVEMBER 1979

FINAL REPORT
CONTRACT NUMBER N60921-78-C-0203
OCTOBER 1, 1978 TO NOVEMBER 31, 1979

Approved for Public Release
Distribution Unlimited

DDC FILE COPY

Prepared for
NAVAL SURFACE WEAPONS CENTER
WHITE OAK, SILVER SPRING, MARYLAND 20910

80 3 6 043

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER 6

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER 9

4. TITLE (and Subtitle)
Aerodynamic Computer Code for Computing Pressure Loads on a Missile at High Supersonic Speeds and Generalized Attitudes.

5. TYPE OF REPORT & PERIOD COVERED
Final Report, [redacted] 1 Oct [redacted]78-[redacted] Nov[redacted]79

6. PERFORMING ORG. REPORT NUMBER
MDC G8223 31

7. AUTHOR(s) 10
Kenneth K. Wang

8. CONTRACT OR GRANT NUMBER(s)
15 N60921-78-C-0203

9. PERFORMING ORGANIZATION NAME AND ADDRESS
McDonnell Douglas Astronautics Company
5301 Bolsa Avenue
Huntington Beach, California 92647

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS
Naval Surface Weapons Center
White Oak
Silver Spring, Maryland 20910

12. REPORT DATE
11 Dec[redacted]79

13. NUMBER OF PAGES
223

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

12 232

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
For public release, distribution unlimited.

14 MDC-G8223

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
Aerodynamics
Pressure Distribution
Structural Analysis
High Supersonic Speeds

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

An aerodynamic computer code was developed for calculating the pressure distribution on a missile or its components, e.g., body and wings and subsequently interpolating the results by surface fit at locations as specified for structural analysis using the NASTRAN computer code. The code is valid for high supersonic speeds and generalized attitudes of the missile.

389310

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

PREFACE

This document reports the work performed by McDonnell Douglas Astronautics Company for the Naval Surface Weapons Center, White Oak Laboratory, Silver Spring, Maryland, for the development of an aerodynamic code for computing pressure loads on a missile and/or its components at high supersonic speeds and generalized attitudes. The work was accomplished under contract number N60921-78-C-0203. The Naval Sea Systems Command sponsored the activity.

Capabilities for analyzing a missile traveling at high supersonic speeds ($2.5 < M_{\infty} < 6$) and generalized attitudes have been incorporated into an existing MDAC aerodynamic code to provide the pressure loads at locations specified by the output of the NSWC computer codes PING and BING for stress analysis by NASTRAN.

The theoretical background of the method for analyzing the high supersonic case is described in this report. The missile attitude has been generalized to include the yaw and roll in addition to the angle of attack. A detailed description of the input parameters is presented together with test cases and operating instructions. A complete listing of the code is included.

Accession For
NTIS GRA&I ☒
DDC TAB □
Unannounced □
Justification______
By______
Distribution/
Availability Codes
Dist A | Avail and/or special

CONTENTS

LIST OF SYMBOLS

C_p	Pressure coefficient, $(P-P_\infty)$ $\frac{1}{2}$ ρ U^2
f_n	Nose finess ratio
M_∞	Free-stream Mach number
x, y, z	Coordinates
α, β, γ	Angle defining missile attitude (see Fig. 1)

SUMMARY

This report represents the work performed under contract number N60921-78-C-0203 for the Naval Surface Weapons Center, White Oak Laboratory, Silver Spring, Maryland, under Naval Sea Systems Command sponsorship for the development of an aerodynamic code for computing pressure loads on a missile at high supersonic speeds and generalized attitudes.

A number of subroutines needed for computing the aerodynamic pressure at high supersonic speeds ($2.5 < M_\infty < 6$) have been developed and incorporated into an existing MDAC aerodynamic code for analyzing the pressure loads on a missile. For treating the missile attitude in the general case, the code was extensively modified to reflect the required changes in the input format, code structure, and output results. After the calculations are made, the code generates the pressure distribution on the missile at the centroid of finite elements as given. Results are produced as punched card output in a NASTRAN bulk-data format.

Section 1
INTRODUCTION

Development of an aerodynamic code (Reference 1) for the calculation of the pressure load on a missile at the centroid of finite elements generated by the NSWC computer code PING and BING (References 2 and 3) was completed in 1978 (contract number N60921-076-C-0164). Because it was based on the numerical solution of a set of linearized governing equations for the fluid flow, its applications are restricted to the subsonic and low-supersonic ($M_\infty < 2.5$) region. With recent improvements in missile technology, it was deemed useful to extend its range of validity to higher supersonic speeds. Additionally, the limitation on missile-attitude changes to angle-of-attack excursions needs to be removed to include the attitude changes in both yaw and roll. The present effort was undertaken to meet these requirements in order to provide the necessary tool for structural analysis.

To minimize the necessary effort, the earlier code was employed as the base into which the high-supersonic and the generalized-missile-attitude capabilities were incorporated. Functions of the original code such as input management, coordinate specification, and transform and output generation were used with a minimum of changes.

Section 2
AERODYNAMIC THEORY - HIGH SUPERSONIC SPEEDS

For high supersonic speeds, the linearized method based on small perturbations for lower Mach numbers ($M_\infty < 2.5$) can no longer be employed for analyzing the pressure loads on missiles. Although exact numerical analyses such as the method of characteristics or finite-difference methods are available, the developmental and computing effort required for their application to the subject problem eliminated them from consideration.

Because it was considered to be the most suitable for the scope of the present effort and for possible future applications, the shock-expansion method was selected for calculating the pressure load on missiles at high supersonic speeds. An explanation of this method follows.

Since the introduction by Epstein (Reference 4) in 1931, the shock-expansion method has been extensively developed by many researchers. The generalized shock-expansion method predicts the pressure distribution for flow about pointed bodies of revolution with reasonable accuracy. In particular, the shock-expansion method (Reference 5) modified for the case where the hypersonic similarity parameter (M_∞/f_n, ratio of free stream Mach number to nose fineness ratio) is close to 1, suits the present application; its coded version (Reference 6) is used with appropriate modifications.

The basic approach and theoretical derivation of the method are described in detail in Reference 5 and will not be repeated here. However, a brief description of the calculational procedure is presented in the following sections. The method starts with the cone solution for bodies of revolution or the oblique-shock solution for two-dimensional flow. For pointed bodies of revolution, the pressure coefficient C_p after the shock and for each subsequent segment is obtained as the sum of two parts. One represents the pressure coefficient at zero incidence (C_{po}), and the other represents

PRECEDING PAGE NOT FILMED
BLANK

the pressure coefficient ($C_{p\alpha}$) due to the incidence α. The zero-incidence pressure coefficient C_{po} is computed using either the second-order theory or that developed in Reference 7, depending on the value of the parameter M_∞/f_n.

The pressure coefficient $C_{p\alpha}$ for the finite incidence case is calculated using the following relations (References 8 and 9):

$$C_{p\alpha} = \left[A_1 T + A_2 T/M_\infty{}^2 - A_3/M_\infty{}^2\right]\left(\frac{\alpha}{\mu}\right)$$

$$+ \left[A_4 T + A_5 T/M_\infty{}^2 + A_6/M_\infty{}^2\right] \left(\frac{\alpha}{\mu}\right)^2 \qquad (1)$$

where

$$A_i = a_{0,i} + a_{1,i} \cos \phi + a_{2,i} \cos 2\phi \qquad (i = 1, 2, \ldots.6)$$

$$T = \sin B\mu \tan\mu$$

The constants B and $a_{j,i}$ are those listed in Reference 9. μ and ϕ are the half-cone angle and the meridian angle, respectively.

The pressure coefficient $C_p = C_{po} + C_{p\alpha}$ obtained at each step of computation is used for calculating the pressure at the end of each segment. As expressed in Reference 5:

$$p_3 = p_c - (p_c - p_2)e^{-\eta} \qquad (2)$$

where

p_3 = pressure at the end of segment.
p_c = pressure of equivalent cone = $\frac{1}{2}\gamma M_\infty{}^2\, p_\infty\, C_p + p_\infty$.
p_2 = pressure result from either 2-D expansion or compression between segments.
η = function of pressure gradient and streamwise distance.

For 2-D flows, either oblique shock or expansion relation is used depending on the flow-deflection angle. The pressure load at the centroid of each panel is calculated by tracing the streamline over the surface of the wing.

Section 3
GENERALIZED MISSILE ATTITUDE

For specifying the attitude of a missile in flight, its pitch, yaw and roll angles are usually required. An aerodynamic analysis limited to pitch-plane motions will be unrealistic and of rather restricted use. Extending the MDAC aerodynamic code to include the effect of yaw and roll expands its range of application.

3.1 COORDINATE SYSTEM SPECIFYING MISSILE ATTITUDES

The coordinate system used for defining the pitch angle α, the yaw angle β, and the roll angle γ of the missile in relation to the freestream-flow vector $\overline{V}$ is shown in Figure 3-1. The successive rotation required to transform from the flow-field coordinate system (X_1, Y_1, Z_1) to that of the missile (X_a, Y_a, Z_a) is described as follows. First, the coordinate (X_1, Y_1, Z_1) is rotated about the Y axis by an angle α and forms the intermediate coordinates (X_2, Y_2, Z_2). Next, it is rotated about the Z_2 axis by an angle β' to form the coordinates (X_3, Y_3, Z_3). Finally, a rotation about the X_3 axis by an angle γ leads to the aerodynamic coordinates (X_a, Y_a, Z_a) for aerodynamic analysis of a missile.

As shown in Figure 3-1, either α, β and γ or α, β' and γ can be used for defining missile attitude. For the present case, the first definition was adopted, i.e., the pitch angle α, the yaw angle β, and the roll angle γ. Note that both α and β are projections of angles in the X_1-Z, and X_1-Y planes, respectively. In other words, they are defined in terms of angle in the flow or inertial-coordinate system. The reason for their selection will become clear when the method for analyzing the pressure loads is discussed.

3.2 EFFECT OF YAW AND ROLL ANGLES ON AERODYANMIC PRESSURE

When a missile is considered as a whole rather than as separate components, accurate analysis of the aerodynamic pressure on its surface in the presence

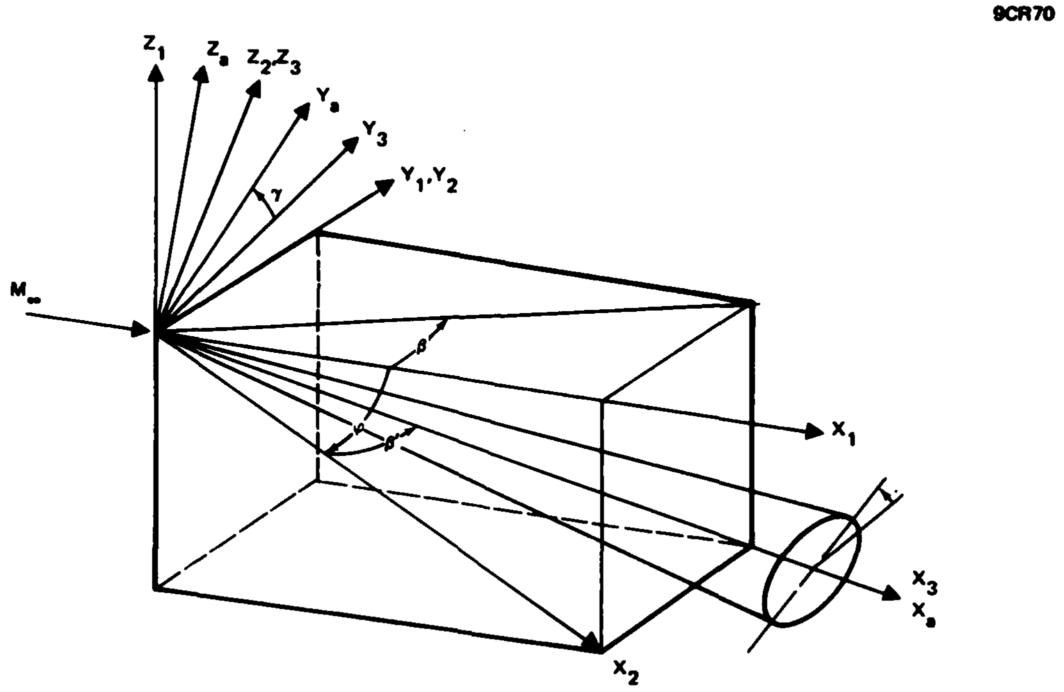


Figure 3-1. Missile Attitude Coordinates

of yaw and/or roll becomes somewhat laborious. A simpler method that provides results adequate for present needs must be developed to reduce the expected effort. For this purpose, a scheme consistent with the basic approach used in analyzing the pressure load was adopted for incorporating the additional effect of yaw and/or roll.

As described in Reference 1, the approach employed is based on the small-perturbation assumption which requires that both angle of attack and yaw be small. This consideration permits the linearization of the governing fluid-dynamic equations and the analysis of the pressure load on the missile as individual components (body and wings). Their interactions, however, are subsequently calculated and added to that of the appropriate component. As described below the effect of yaw and roll on the aerodynamic load on the missile is similarly analyzed taking advantage of this approach.

The combined effect of angle of attack α and yaw β on the body can be viewed as that of an equivalent angle of attack α' with a rotation of the body about its axis by an angle γ'. From the geometry, one has

$$\alpha' = \text{arc cos}\left[1.0/\sqrt{1 + \tan^2\alpha + \tan^2\beta}\right] \tag{3}$$

$$\gamma' = \text{arc tan}\left[\tan\beta/\tan\alpha\right] \tag{4}$$

As defined, the effect of roll γ represents only a shift in the angular distribution of the pressure load on the body and is incorporated by substituting the algebraic sum of γ and γ' or γ'' for γ' in the analysis.

The combined effect of angle of attack α, yaw β and the roll γ on the wing pressure load is incorporated by an apparent angle of attack α'' which is

$$\alpha'' = \text{arc sin}\left[\sin\gamma'' \sin\beta' \cos\alpha + \cos\gamma'' \sin\alpha\right] \tag{5}$$

where

$$\beta' = \text{arc cos}\left[\sqrt{\left(1 + \tan^2\alpha\right) / \left(1 + \tan^2\alpha + \tan^2\beta\right)}\right] \tag{6}$$

The advantage of the present approach for incorporating the effect of yaw and roll is that it allows maximum use of the output of the existing analysis and consequently minimizes the effort required to develop a more versatile code. The practical necessity of code reorganization, core-storage rearrangement, and appropriate assembly of additional subroutines have, however, been time consuming, and repeated computer runs were required for program checkout.

Section 4
COMPUTER PROGRAM

The MDAC aerodynamic code (Reference 1) previously developed for NSWC under the contract number N60921-76-C-0164 has been extensively modified to incorporate the high-supersonic-speed and the generalized missile-attitude capabilities. Using the output of NSWC computer codes PING and BING which specify the finite-element and grid-point distribution on the missile, the present code calculates the desired pressure load at the centroid of each element. At the user's option, the pressure load may be generated as punched-card output in the bulk-data format of NASTRAN for subsequent structural analysis.

4.1 CODE CAPABILITIES

The present code as modified is capable of calculating the pressure load for a missile or its components (body or wing) at either subsonic or supersonic speeds. The pitch-plane restriction on its attitude has been removed to allow both yaw and roll in addition to pitch-plane motions. Due to the small-disturbance assumption, the code is valid for slender bodies of circular or near circular cross-section at small angle of attack or yaw.

The code requires a field length of 150, 000 when used on the CDC CYBER 174 computer with KRONOS 2.1 system. For subsonic or low-supersonic cases, approximately 3 minutes computing time is required. For higher supersonic ($M > 2.5$) cases, much less computing time ($<<$ 1.0 min.) is required.

4.2 INPUT DESCRIPTION

Input parameters for numerical computation have been modified as needed, based on the MDAC aerodynamic code previously developed. Only minor changes were required for specifying either the high-supersonic-speed case or the generalized-attitude case. However, for user's convenience, a complete list of input parameters has been included in the following description:

PRECEDING PAGE NOT FILMED
BLANK

XMACH Mach number of flow. Higher supersonic routine employed when greater than 2.5, default value = 2.0.

PINF Free stream pressure, default value = 14.7 psi.

DADEG Incremental angle of attack of missile in degrees, default value = 0.

SID Case identification number for each missile component (up to 20 allowed); default values have been set at 1. Also used for selecting the sign convention for pressure output. Code uses the accepted rule for pressure, i.e., positive acting toward the surface and negative acting away from the surface. For opposite-sign convention, set SID(L) as negative for pressure output on the desired component L (in the order of body, wing surfaces, upper or lower).

YAW Angle of yaw in degrees with respect to the flow.

BRØLL Missile roll about body axis (degree).

The following input parameters are required for the case where missile body is included:

NBPFL Number of body profile stations for specifying body geometry inputs $r = r(x)$; up to 51 stations allowed.

XB ≤ 51 Axial coordinate of body stations in the direction of flow for specifying the body profile.

RB ≤ 51 Radial coordinate of body profile at corresponding body stations.

ZD ≤ 51 Body camber at corresponding body stations.

ARB Angle of attack of missile body in degrees, default value = 0.

As is similar to the overall organization of the code, the input parameters and data have been organized into two main groups in accordance with their assigned functions. The first group contains all the parameters needed for the specification of case selection, missile configuration, and flow conditions. They form the input parameter of the namelist UNIFID. The second group specifies the finite-element and grid-point system as generated by the NSWC computer codes BING and/or PING. It includes the specification for the coordinate system used for the finite elements.

For the convenience of the user, card format for inputs are described in their sequential order as required by the code.

A. Aerodynamic Control Parameters.

Card 1 Title and case description, format (20A4), alphanumeric.

Card 2 $UNIFID, begins at Column 2 with the symbol $.

Card 3 And additional cards if necessary, contain the following as input:

ICASE = 1 Wing only.

= 2 Body only.

= 3 Complete missile.

NTRANS Number of coordinate transformations required to convert the coordinates for the finite element to the coordinates used in the aerodynamic code. Two transformations are allowed for each missile component, in the following order; body, wings, and nacelles.

IPUNCH = 0 No punched card output, default value.

= 1 Punched card output requested.

IRW = 0 Computation results are not saved on tape, default value.

= 1 Aerodynamic computation results are saved on tape 12 for restart, first run only.

= 2 Restart run, bypass all aerodynamic computations and begins computation at the start of pressure interpolation.

POLAR Number of incremental angles of attack, default value = 0.

The following inputs are required for the case where the missile wing is included:

ISØLID = 0 Wing of built-up construction, pressure load on both upper and lower wing surface generated, default value.

= 1 Wing of solid section, sum of the pressure loads on wing surface generated.

IFØRM = 0 Wing of built-up construction, both upper and lower surface height (ZWI) are given at identical locations (XWI, YWI).

= 1 Wing of built-up construction, upper and lower surface height (ZWI) are not given at identical locations.

= 2 Flat wing surface.

NWPI Number of coordinate points specify the wing surface contour (≤30). Code allows a maximum of 20 wing surfaces (upper and lower). Each wing surface must be defined in the form of equation of a surface, ZWI = f(XWI, YWI). A maximum of 30 points are permitted with the provision that the first four points are restricted to the specification of the corners of the wing only. The coordinates for the corners must be given in the following order. Starting with the innermost point of the leading edge, the remaining corners are to be given in a clockwise order viewing from the above.

NWING Number of wings or tails. A maximum of 10 is allowed, default value = 0.

XWI(i, j), YWI(i, j), ZWI(i, j) Wing-surface coordinates where indices i and j designate the surface coordinate point and wing surface, respectively.

DIHED Dihedral angle, degrees.

PIVØT Indicates dihedral, 0 for no dihedral, and >0 for dihedral present.

ARW Angle of attack of wing in degrees, with respect to the body axis if complete missile is considered, default value = 0.

The following input parameters are required for the complete-missile case (ICASE = 3) except for high-supersonic-speed (M > 2.5) case:

NWPANL   Number of wings directly connected to the body.

XLG   Axial distance at the intersection of wing leading edge and body for each wing.

XTG   Axial distance at the intersection of wing trailing edge and body for each wing.

Card 4 $END   Begin at column 2 the symbol $ for ending input.

B. Finite-Element and Grid Inputs. The second group of input cards are arranged immediately following Card 4, as follows:

Cards 5 and 6   Coordinate-system specifications in the format of NASTRAN bulk-data deck [page 2.4-49 - 2.4-54, NASTRAN User's Manual, NAS SP-221(01), 1972]. Three position vectors — A, B, and C — are used to define the coordinate system. The first defines the origin, the second defines the Z axis, and the third defines a point in the XZ plane.

For Card 5 the following format is used:

Col 1-8, Coordinate system CØRDJRbb for rectangular, CØRDJCbb for cylindrical, and CØRDJSbb for spherical, where J identifies the coordinate system numbers.

Col 9-16 Coordinate identification number J, integer.

Col 17-24, Reference coordinate system, integer, optional.

Col 25-32, 33-40, 41-48, Components of vector A (3F8.2).

Col 49-56, 57-64, 65-72, Components of vector B (3F8.2).

For Card 6:

Col 1-8, blank.

Col 9-16, 17-24, 25-32, Components of vector C (3F8.2).

Cards 5 and 6 are to be repeated as many times as there are coordinate systems (NTRANS) required for specifying the finite element. They are to be arranged in the order of body and wings.

The finite element and grid specifications are arranged immediately following the coordinate-system input in the same order, i.e., body, wings.

Card 7: Specifies the number of grid cards (NGRIDP) and element cards (NE), Format (2I10).

The grid and the element cards as generated by the NSWC computer-code BING are sorted into two separate groups. They are then placed immediately after card 7 with the grid cards in front.

Some remarks regarding the input preparation need to be made here to facilitate the use of the aero code.

For the body-only case, inputs of the body profiles, expressed as r = r(x), for aero-panel generation must be in the aero-coordinate system. It is required that x be the axial distance from the tip and r the radius of the body cross section (Figure 4-1). For the finite element and grid point, their relationship to the basic coordinate system and the basic coordinate system to aero-coordinate system are to be specified in the coordinate input card immediately following the namelist input. The grid card and the element card are to be separated and placed after the coordinate system specification, with a card specifying their numbers placed behind the coordinate card.

The input for the wing-only case requires the specification of both the wing planform and its surface contour. The input for the description of the wing planform requires the coordinates of corner points and break points. For each individual wing, inputs for those of quadrilateral shape are their four corner coordinates. The codes have allocated the first four values of the coordinates XWI, YWI and ZWI for each wing or wing region, arranged in a counterclockwise order starting from the leading-edge corner of break point. In the case of triangular wings, there are only three corner points. To satisfy the same requirement, the apex or wing tip is to be considered as two coincident points of the same coordinate. The inputs are to be arranged like those of

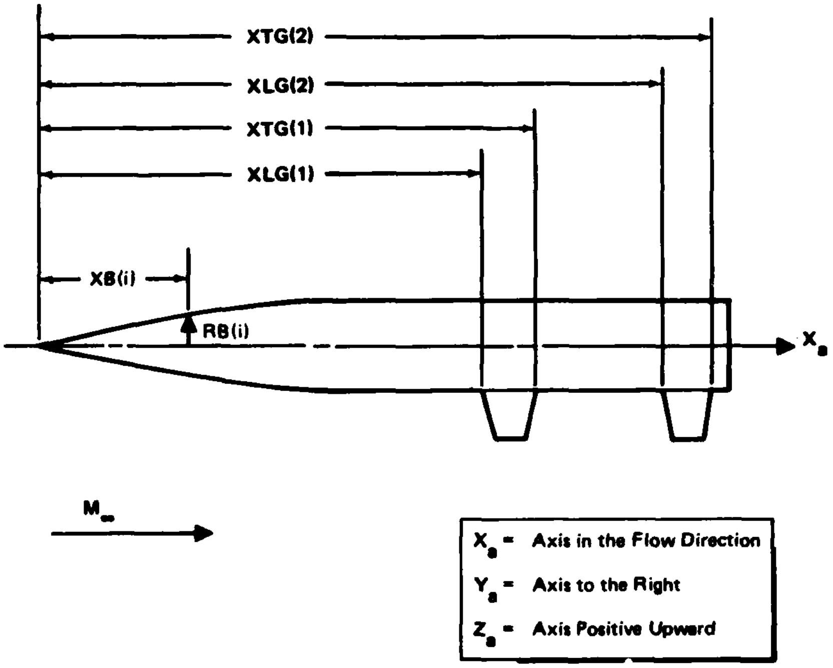

XTG(2)
XLG(2)
XTG(1)
XLG(1)
XB(i)
RB(i)
X_s
M_∞
X_s = Axis in the Flow Direction
Y_s = Axis to the Right
Z_s = Axis Positive Upward

Figure 4-1. Body Profile Input

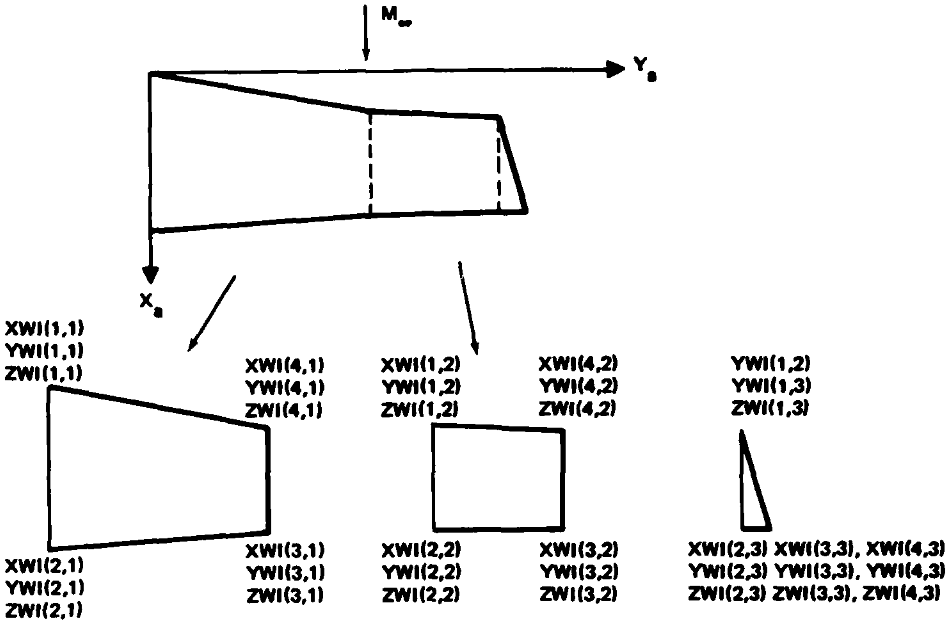
the quadrilateral shape, with leading-edge body-corner point as the starting point. Wings with complex shapes (see Figure 4-2) must be subdivided into appropriate quadrilateral or triangular regions and the inputs prepared accordingly. The only requirement is that the control chords, apart from either the leading edge or the trailing edge, must be in the streamwise direction.

All inputs for the wing surface are to be given in the form of the equation of surface Z = f(x, y) in terms of XWI, YWI and ZWI. Starting with the fifth value, XWI (5, I), YWI(5, I), ZWI(5, I), their total number may not exceed 26 points. For wings with flat surfaces, only the first four points are necessary.

Inputs defining wing thickness, camber and twist have been eliminated to reduce the input preparation effort. The code generates the required values of camber, thickness slope, and twist angle for the wing using the existing surface-fit scheme from the input.

Similar to the body-only case, the finite-element and grid-point coordinate-system cards are placed immediately behind the namelist input. They are

9CR70



NWING = 3, NTRANS(1) = 1, 1, 1,
If Built-Up Wing is Used, NWPI(1), NWPI(2), and NWPI(3) Must be Specified

Figure 4-2. Subdivision of Complex Wing

followed by a card specifying the number of grid cards and element cards. The grid-point and element card as generated by PING occupies the last position in the input.

The coordinate system used for the wing can be any one of three systems, i.e., rectilinear (x, y, z), cylindrical (r, θ, z) and spherical (r, θ, ϕ). In terms of the input parameters, they are represented by XWI, YWI and ZWI, respectively, for all three systems.

For the complete-missile case, the input can be considered as a combination of these two cases, i.e., the body-only and the wing-only cases. Only the parameters ICASE and NTRANS need be corrected to reflect the changes.

Both coordinate-system cards and finite-element cards are to be grouped together in the order of body, wings, and nacelles.

As a result of the approach used for the analysis, the computations for the high-supersonic-flow case are considerably simpler. Accordingly, the code treats the complete missile as a rule.

Section 5
TEST CASES

Test cases were selected for aerodynamic analysis by the present code for the purpose of verifying the theoretical approach and code modifications. In Reference 10, the experimental results from tests conducted for a model with delta planform wings and cylindrical body at supersonic speeds(M_∞= 2.00 and 2.58) are available and suitable for our purposes. The model configuration with dimensions is shown in Figure 5-1. Location of the pressure taps for wing and body are shown in Figures 5-2 and 5-3, respectively. Tests were made for angles of attack and yaw varying from 0° to 5°.

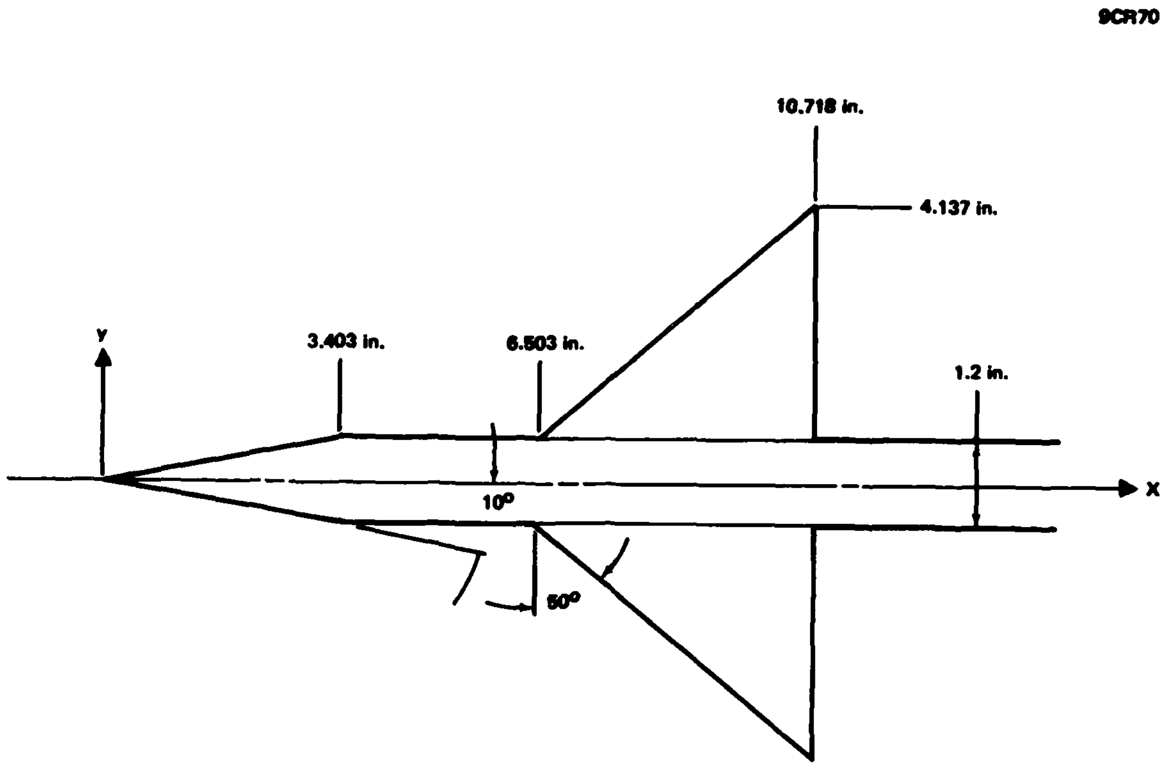


Figure 5-1. Test Model (Reference 10)

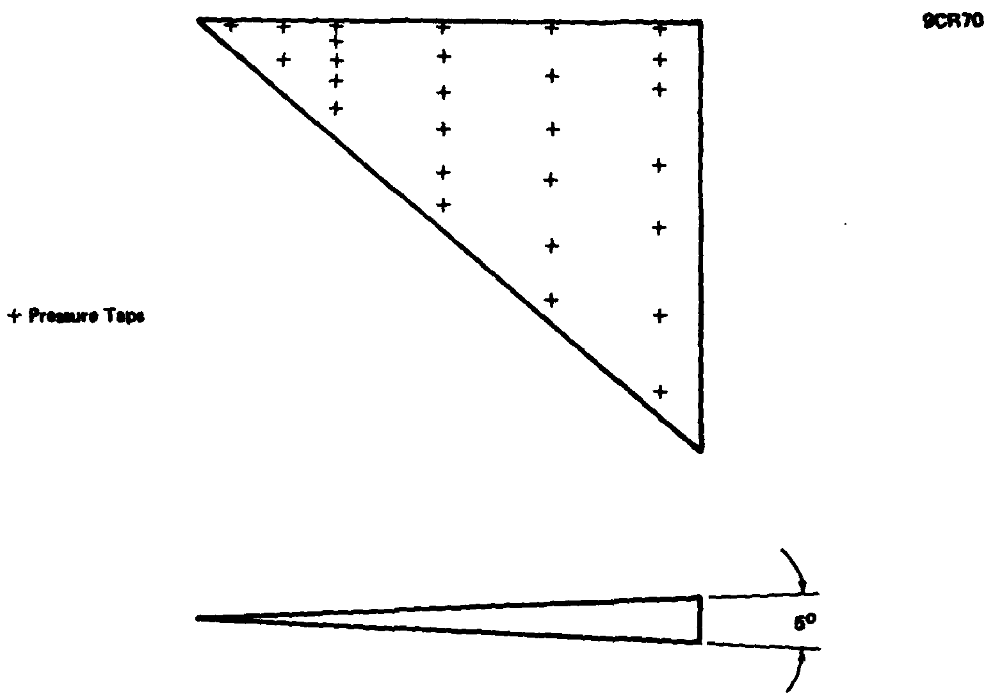


Fig. 5-2. Wing Plan Form and Pressure Tap Locations

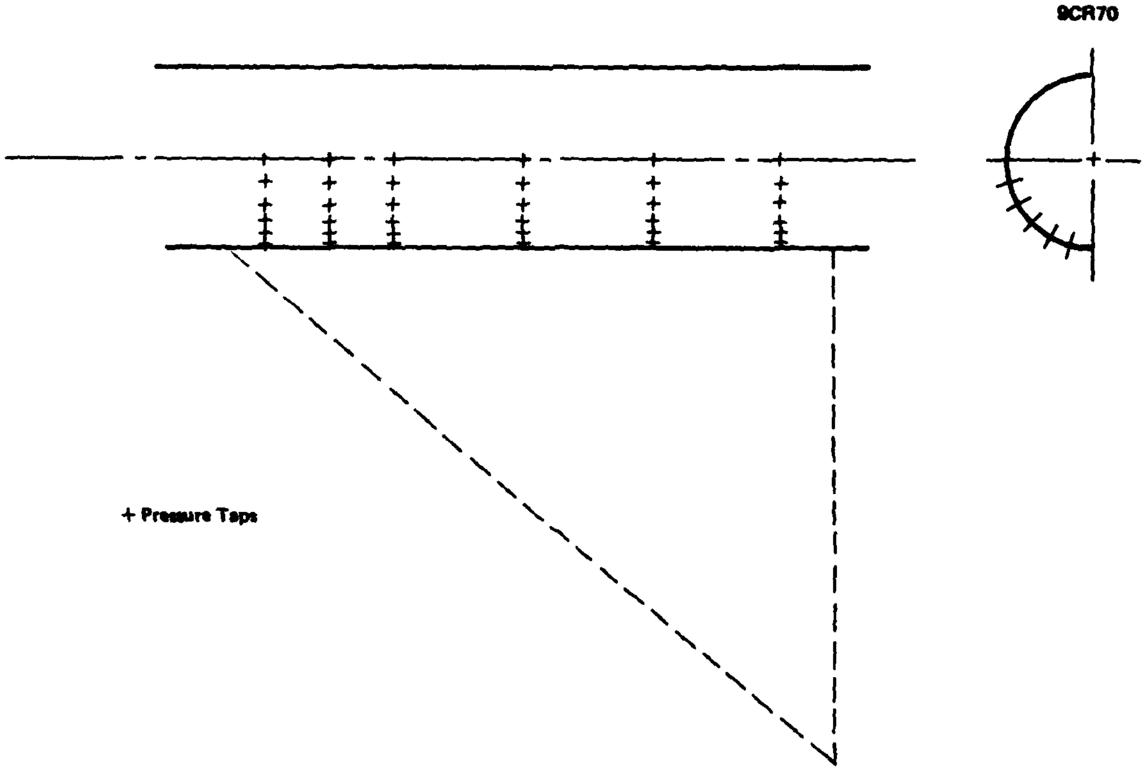


Fig. 5-3. Missile Body and Pressure Tap Locations

To demonstrate the ability of the code in analyzing the pressure load on a missile at high-supersonic speed and generalized attitude, computations were made for missiles flown at (a) combined angle of attack (5°) and yaw (5°) at Mach 2.0 and, (b) high-supersonic case at Mach 2.58 and 5° angle of attack.

5.1 COMBINED ANGLE OF ATTACK AND YAW

The following is a list of input cards, in sequence:

(I) Title Card.

Descriptive title of the case.

FORMAT: (20A4).

(II) Namelist UNIFID input.

Aerodynamic control parameters and geometries of missile.

FORMAT: Real or integer in accordance to the variable.

Test Case:

ICASE = 3	Complete missile.
NTRAN(1)=1, 1	No intermediate coordinate system for wing or body.
ISØLID = 1	Solid wing, sum of upper and lower pressure.
NBPFL = 15	Number of input points for body profile.
NWPI(1)=4	Number of input points for wing surface.
NWING=1	One wing section.
IFØRM(1)=2	Flat-wing surface.
ARB = 5.0	Angle of attack of missile.
YAW = 5.0	Angle of yaw of missile.
XMACH = 2.0	Flow Mach number.
NWPANL = 1	Number of wings directly connected to the body.
XLG(1)=6.053	Axial distance of the inner leading edge.
XTG(1)=10.718	Axial distance of the inner trailing edge.

For body profile:

XB(1)=0.0, 0.5, 1.0, 1.5, 2.0,....

RB(1)=0.0, 0.0881, 0.1763, 0.2664, 0.3526,....

For the wing surface, only four points are needed for a solid wing:

XWI(1, 1)=6.503, 10.718, 10.718, 10.718,

YWI(1, 1)=0.6, 0.6, 4.137, 4.137,

ZWI(1, 1)=0.0, 0.184, 0.0, 0.184,

(III) Coordinate Card.

Notes: The coordinate cards are needed for input of vectors A, B, and C which are used in defining the relation between the coordinate systems of the finite element (X, Y, Z) and the aero codes (X_a, Y_a, Z_a). In accordance with NASTRAN, vector A defines the origin, vector B defines the Z axis, and vector C defines the X-Z plane.

FORMAT:

NASTRAN bulk-data deck, first card (A8, 2I8, 6F8.2) and second card (A8, 3F8.2).

Test Case:

Two sets of coordinate cards are required, one for the body and one for the wing. The body finite-element coordinate system has its origin coincident with the aero code and its Z axis coincident with the X axis of the aero code as shown in the sketch. Hence, the vector A has component (0.0, 0.0, 0.0), vector B has component)(1.0, 0.0, 0.0), and vector C has component (1.0, 0.0, 1.0).

The finite-element coordinate system for the wing, as shown in the sketch, requires the input of vector A with component (0.0, 0.0, 0.0), vector B (0.0, 0.0, 1.0), and vector C (1.0, 0.0, 1.0).

(IV) Finite-Element Cards.

Notes: The finite-element and grid-point cards as generated by BING and PING are used for defining element coordinates. They must be divided into two separate groups with the grid-point card preceding the element card. A card specifies that the numbers of grid card and element card in a format (2I10) must be placed before them.

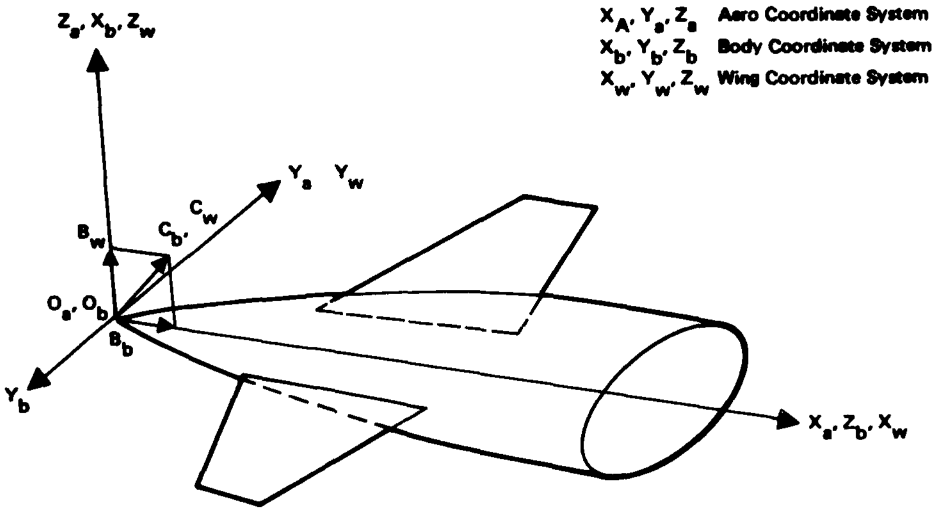
FORMAT:

For grid and element number card (2I 10), grid and finite-element cards must be the same as generated by BING or PING.

Test Case:

As shown in the input listing, first set of finite-element grid points and element cards are that of the body and the subsequent set is for the wing surface.

9CR70


X_A, Y_a, Z_a Aero Coordinate System
X_b, Y_b, Z_b Body Coordinate System
X_w, Y_w, Z_w Wing Coordinate System
Z_a, X_b, Z_w
Y_a Y_w
C_w
$C_b,$
B_w
O_a, O_b
B_b
Y_b
X_a, Z_b, X_w

Results of Computations

Using the inputs as described, computations were made for the pressure load on the body and wing of the missile tested in Reference 10. In Figure 5-4, the pressure load on the body at the meridian angles of 45° and 135° were compared with that of the experiment. Good correlation with the experimental measurements was obtained. In Figure 5-5, comparisons with the experiment were made for the pressure load on the wing and the results were satisfactory.

It is seen that the code can be used for analyzing the aerodynamic loads on missile at attitudes that may include yaw and roll. Its applicability is therefore considerably extended.

5.2 HIGH SUPERSONIC SPEEDS

The input for this case differs from the previous case only in the Mach number, i.e., 2.58 instead of 2.0. All the remaining input parameters have been retained for this computation and hence will not be repeated here.

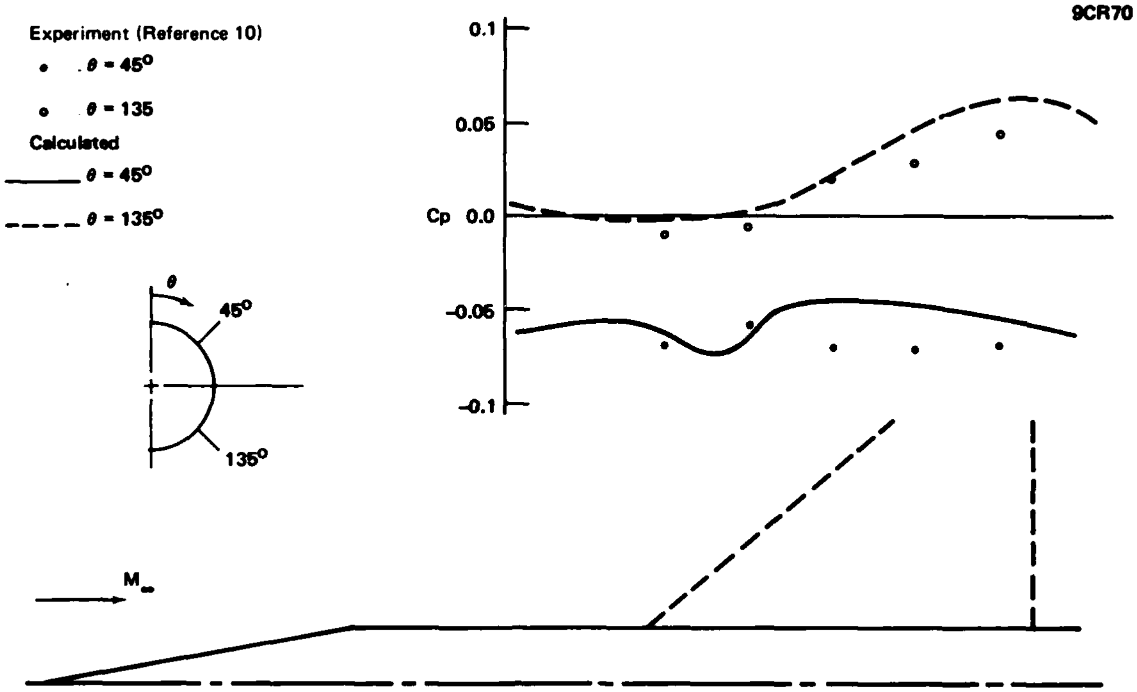


Fig. 5-4. Pressure Distribution on Body at Mach 2.0, 5° Angle of Attack and 5° Yaw

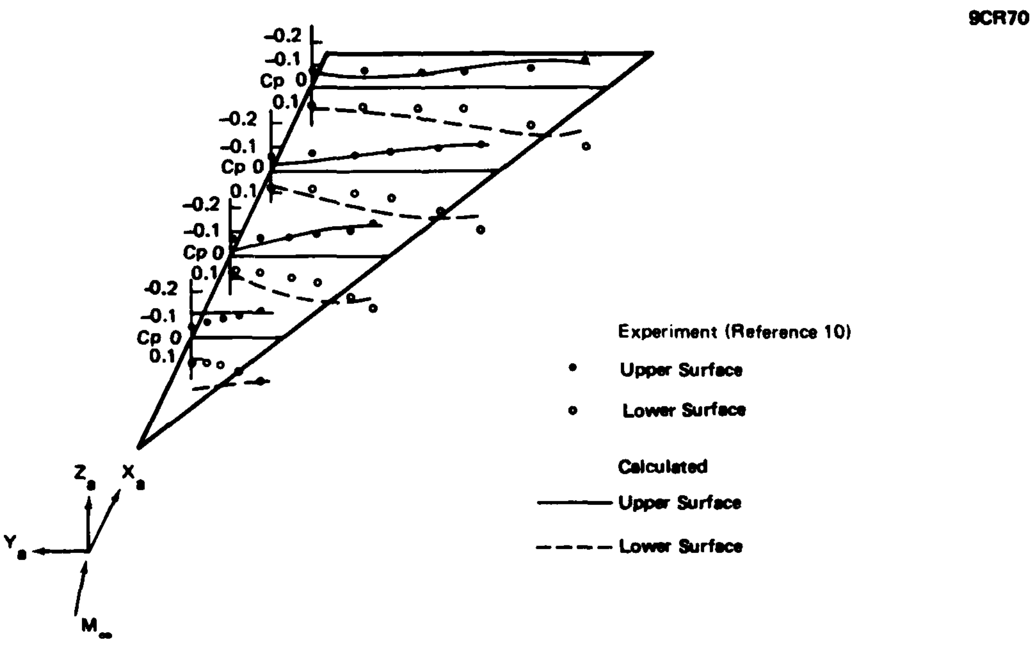


Figure 5-5. Pressure Distribution on Wing at Mach 2.0, 5° Angle of Attack and 5° Yaw

Computational Results

Results of computations for flow at Mach 2.58, illustrated in Figure 5-6 for the body and Figure 5-7 for the wing, show that the pressure distribution on the body correlates poorly with the test data. It is believed that this results from a substantial effect of the wing on the body. This effect should diminish with increasing Mach number. The pressure distribution on the wing surface appears to be acceptable over most of the wing surface.

It should be pointed out that due to the scarcity of experimental data for pressure distributions on complete missiles, the test case described was computed for a Mach number value slightly above the lower limit for the high-supersonic-speed analysis. However, for bodies of simple geometries (circular cones and triangular wing), the test cases were computed at Mach numbers 6.85 and 5.05, and the results correlated rather convincingly. In Figure 5-8, pressure coefficients on circular cone with half-cone angle of 12.5° at angles of attack of 3°, 6°, and 12° were calculated, and the results were very good when compared with available data. In Figures 5-9 and 5-10, pressure coefficients and shock shape for an ogive at an .e of attack were

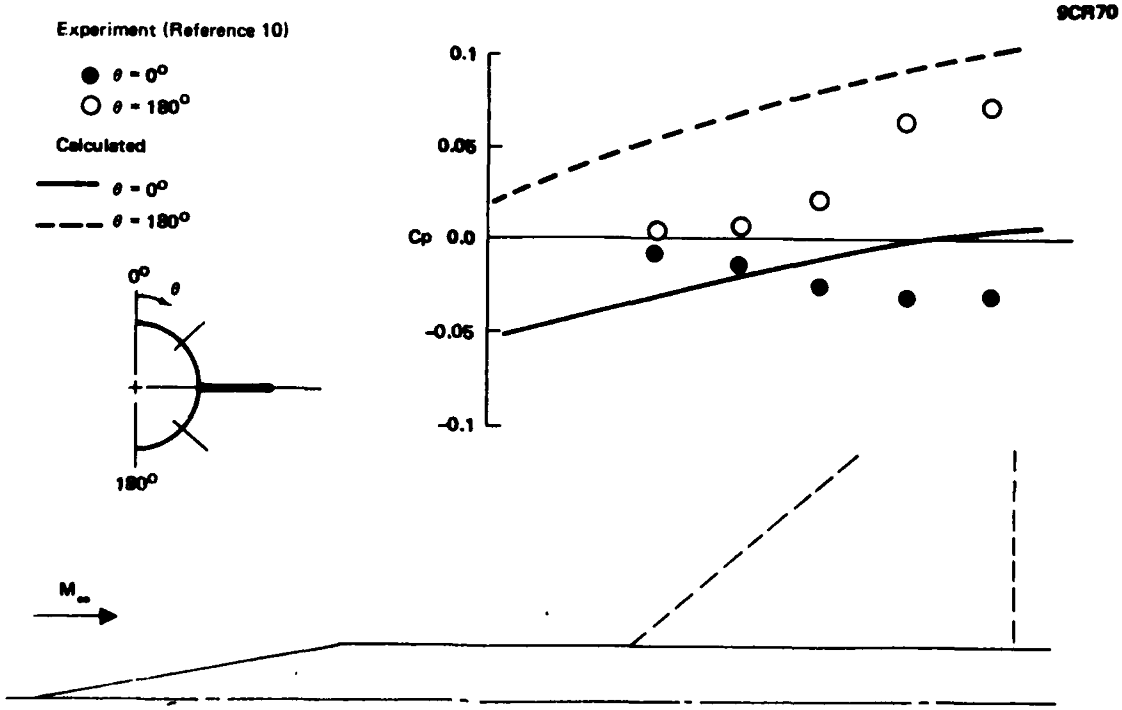


Figure 5-6. Pressure Distribution on Body at Mach 2.58, 5° Angle of Attack

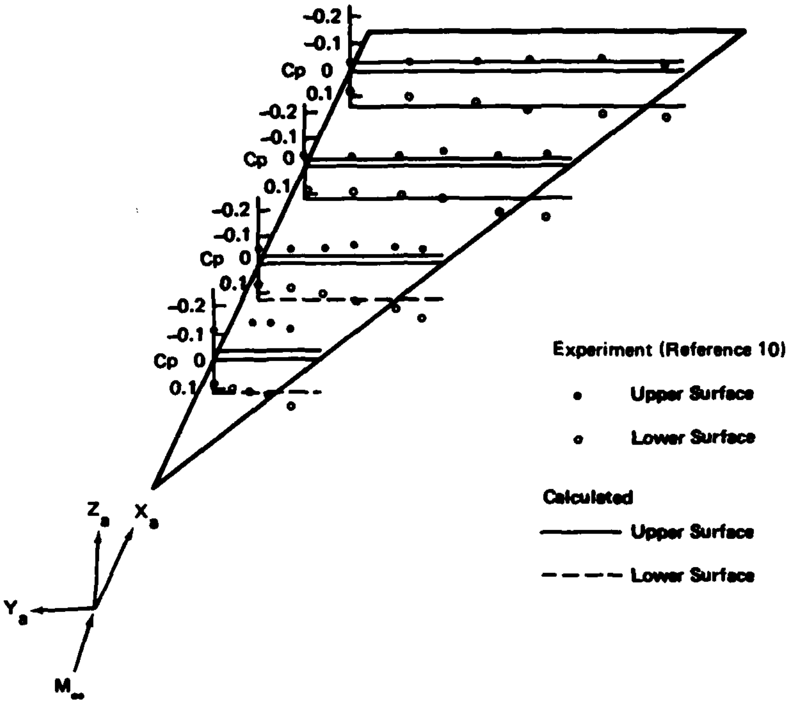


Figure 5-7. Pressure Distribution on Wing at Mach 2.58, 5° Angle of Attack

computed, and the results were satisfactory as shown. In Figure 5-11, the pressure on a 75° sweep, triangular wing was calculated using the shock-expansion method. Comparisons with data show good correlation.

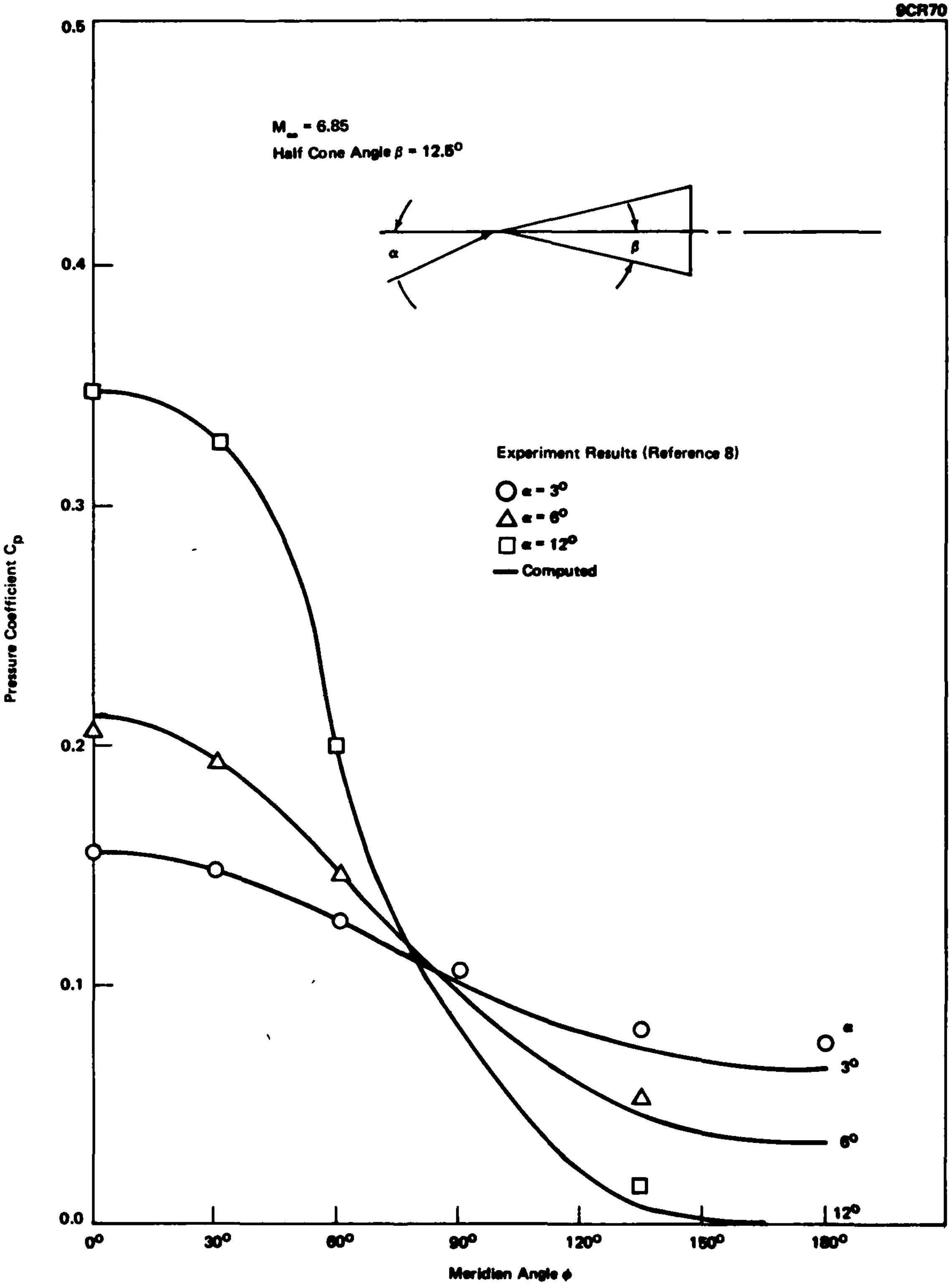


Figure 5-8. Pressure Coefficient for Cone

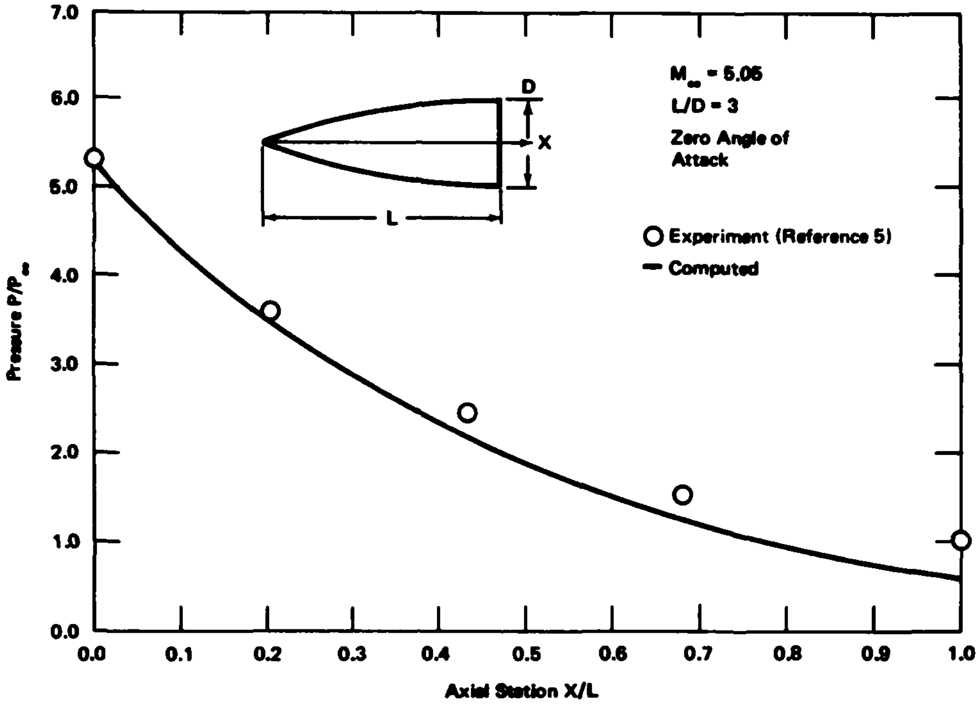


Figure 5-9. Pressure Distribution for Ogive

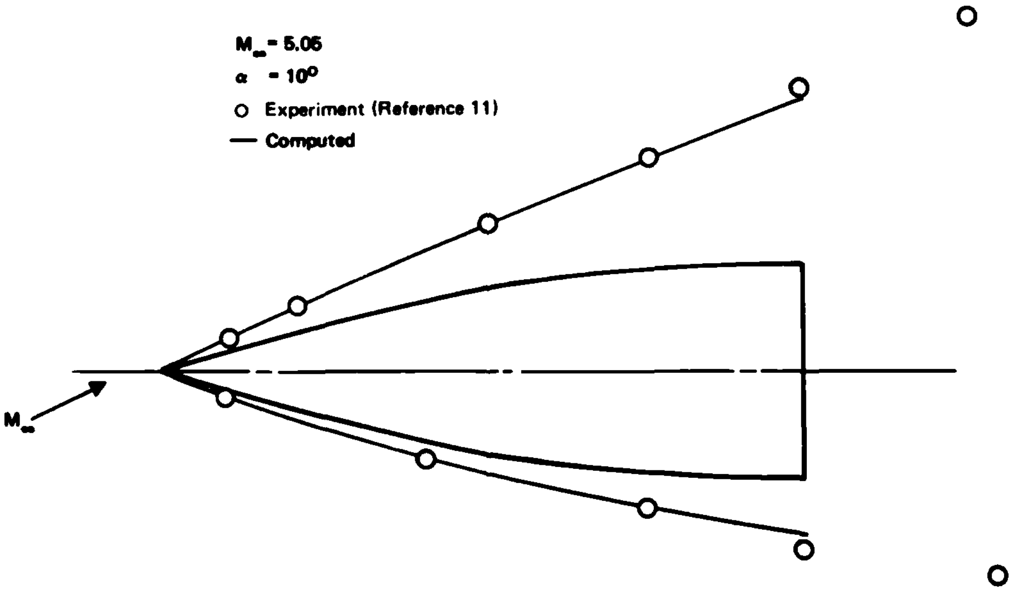


Figure 5-10. Shock-Wave Shape for Ogive

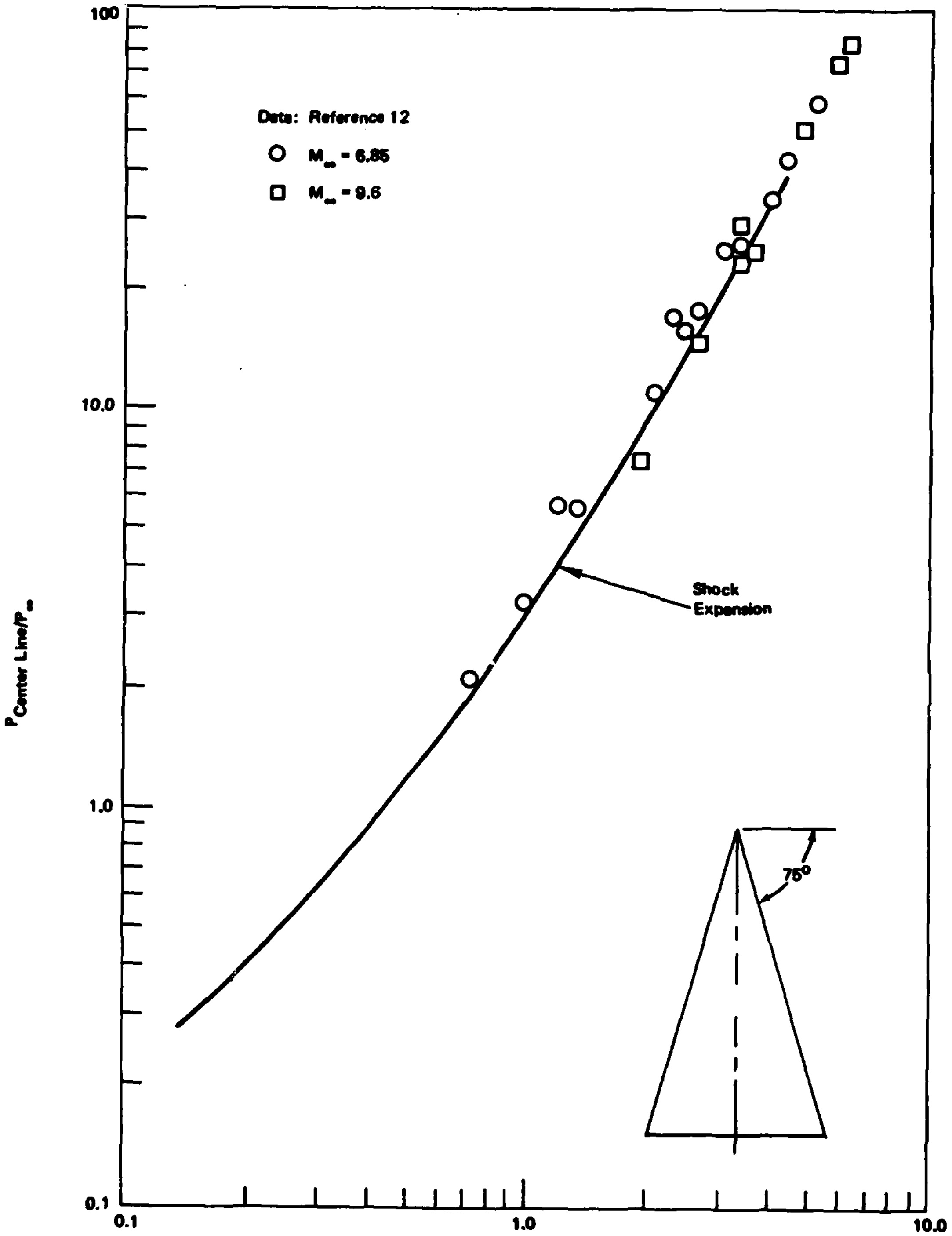


Figure 5-11. Delta Wing Center Line Pressure

5.3 TEST-CASE INPUT CARDS

```
    EXPERIMENT CASE FOR CHECKING GENERALIZED MISSILE ATTITUDE MODE
 SUNIFID
   ICASE=3, NTRANS(1)= 1, 1, NBPFL= 15, NWPI(1) =  4, NWING = 1,
    ARB= 5.0, XMACH = 2.00, YAW= 5.0,  BROLL = 0.0, IFORM(1) =2,
    THICK = 1.0, NWPANL = 1, PINF = 14.7, XLG(1) = 6.503, XTG(1) =10.718,
    ISOLID = 1,
    XB(1) = 0.0, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.3, 3.403, 3.5, 5.5, 7.5,
         9.5, 12.0, 15.0,
    RB(1) = 0.0, 0.0881, 0.1763, 0.2664, 0.3526, 0.4408, 0.5289, 0.5818,
         6*0.6,
    XWI(1,1) = 6.503, 3*10.718,
    YWI(1,1) = 2*0.6, 2*4.137,
    ZWI(1,1) = 0.0, 0.184, 0.0, 0.184,
 $END
CORD2C         1       0     0.0     0.0     0.0     1.0     0.0     0.0
             1.0     0.0     1.0
CORD2R         3       0     0.0     0.0     0.0     0.0     0.0     1.0
             1.0     0.0     1.0
        13        11
GRID           1       0   0.000   0.000   0.000       0       4
GRID           2       1    .071   0.000    .403       1       4
GRID           3       1    .071  30.000    .403       1       4
GRID           4       1    .071  60.000    .403       1       4
GRID           5       1    .071  90.000    .403       1       4
GRID           6       1    .071 120.000    .403       1       4
GRID           7       1    .071 150.000    .403       1       4
GRID           8       1    .071 180.000    .403       1       4
GRID           9       1    .071 210.000    .403       1       4
GRID          10       1    .071 240.000    .403       1       4
GRID          11       1    .071 270.000    .403       1       4
GRID          12       1    .071 300.000    .403       1       4
GRID          13       1    .071 330.000    .403       1       4
CTRIA2         1       1       3       2       1   0.000
CTRIA2         2       2       4       3       1   0.000
CTRIA2         3       3       5       4       1   0.000
CTRIA2         4       4       6       5       1   0.000
CTRIA2         5       5       7       6       1   0.000
CTRIA2         6       6       8       7       1   0.000
CTRIA2         7       7       9       8       1   0.000
CTRIA2         8       8      10       9       1   0.000
CTRIA2         9       9      11      10       1   0.000
CTRIA2        10      10      12      11       1   0.000
CTRIA2        11      11      13      12       1   0.000
        18        11
GRID          83       3  6.5030     0.6     0.0       3
GRID          95       3  7.2055     0.6     0.0       3
GRID         107       3  7.9080     0.6     0.0       3
GRID         119       3  8.6110     0.6     0.0       3
GRID         131       3  9.3130     0.6     0.0       3
GRID         143       3  10.016     0.6     0.0       3
GRID         155       3  10.718     0.6     0.0       3
GRID         211       3  7.2055  1.1895  0.0000       3
GRID         221       3  7.9079  1.1895  0.0000       3
GRID         231       3  8.6105  1.1895  0.0000       3
GRID         241       3  9.3130  1.1895  0.0000       3
GRID         251       3 10.0155  1.1895  0.0000       3
GRID         261       3 10.7180  1.1895  0.0000       3
GRID         222       3  7.9080  1.7790  0.0000       3
GRID         232       3  8.6105  1.7790  0.0000       3
GRID         242       3  9.3130  1.7790  0.0000       3
GRID         252       3 10.0155  1.7790  0.0000       3
GRID         262       3 10.7180  1.7790  0.0000       3
CTRIA2       400     400      83      95     211  0.0000
CQUAD2       401     401      95     107     221     211  0.0000
CQUAD2       402     402     107     119     231     221  0.0000
```

```
CQUAD2      403     403     119     131     241     231 0.0000
CQUAD2      404     404     131     143     251     241 0.0000
CQUAD2      405     405     143     155     261     251 0.0000
CTRIA2      406     406     211     221     222  0.0000
CQUAD2      407     407     221     231     232     222 0.0000
CQUAD2      408     408     231     241     242     232 0.0000
CQUAD2      409     409     241     251     252     242 0.0000
CQUAD2      410     410     251     261     262     252 0.0000
```

5.4 TEST-CASE OUTPUT

5.4.1 Combined Angle of Attack and Yaw

EXPERIMENT CASE FOR CHECKING GENERALIZED MISSILE ATTITUDE MODE

```
SUNIFID

ICASE   =  3,

NTRANS  =  1,  1,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,    0,  0,  0,

IPUNCH  =  0,

IRW     =  0,

POLAR   =  0.0,

ISOLID  =  1,

SID     =  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,
           0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,  0.1E+01,

NBPFL   =  15,

XB      =  0.0,  0.5E+00,  0.1E+01,  0.15E+01,  0.2E+01,  0.25E+01,  0.3E+01,    0.33E+01,  0.3403E+01,  0.35E+01,
           0.55E+01,  0.75E+01,  0.95E+01,  0.12E+02,  0.15E+02,  0.0,  0.0,     0.0,  0.0,  0.0,  0.0,  0.0,

RB      =  0.0,  0.881E-01,  0.1743E+00,  0.2664E+00,  0.3526E+00,  0.4408E+00,  0.5289E+00,  0.5818E+00,  0.6E+00,
           0.6E+00,  0.6E+00,  0.6E+00,  0.6E+00,  0.6E+00,  0.0,  0.0,  0.0,    0.0,  0.0,  0.0,  0.0,  0.0,

ZB      =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,      0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,      0.0,  0.0,  0.0,  0.0,  0.0,

RP      =  0.0,

NWPI    =  4,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,    0,  0,  0,

YAW     =  0.5E+01,

BROLL   =  0.0,

NWING   =  1,

IFORM   =  2,  0,  0,  0,  0,  0,  0,  0,  0,  0,

PIVOT   =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

THICK   =  0.1E+01,

DINED   =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

XWI     =  0.6503E+01,  0.10718E+02,  0.10718E+02,  0.10718E+02,  0.0,  0.0,     0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,      0.0,  0.0,  0.0,  0.0,  0.0,

YWI     =  0.6E+00,  0.6E+00,  0.4137E+01,  0.4137E+01,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,      0.0,  0.0,  0.0,  0.0,  0.0,

ZWI     =  0.0,  0.184E+00,  0.0,  0.184E+00,  0.0,  0.0,  0.0,  0.0,  0.0,      0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,      0.0,  0.0,  0.0,  0.0,  0.0,

XNACEL  =  0.0,

NACELP  =  0,

XN      =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,      0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,      0.0,  0.0,  0.0,  0.0,  0.0,

RN      =  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,      0.0,  0.0,  0.0,  0.0,  0.0,
           0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,      0.0,  0.0,  0.0,  0.0,  0.0,
ZN     = 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,    0.0, 0.0, 0.0, 0.0, 0.0,
         0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,    0.0, 0.0, 0.0, 0.0, 0.0,
NWPANL = 1,
XLG    = 0.6503E+01, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
XTG    = 0.10718E+02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
XMACH  = 0.2E+01,
PINF   = 0.147E+02,
DADEG  = 0.0,
ARB    = 0.5E+01,
ARW    = 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
ARN    = 0.0, 0.0,
$END
```

BODY PANEL CORNER POINT LOCATIONS EXPRESSED IN THE AERO COORDINATE SYSTEM
1 AND 2 INDICATE BODY PANEL LEADING-EDGE POINTS, 3 AND 4 INDICATE TRAILING-EDGE POINTS

PANEL NO	PARTS	X_1	Y_1	Z_1	X_2	Y_2	Z_2	X_3	Y_3	Z_3	X_4	Y_4	Z_4
1	1	5.660	0.000	.600	5.660	.300	.520	6.503	0.000	.600	6.503	.300	.520
2	1	6.503	0.000	.600	6.503	.300	.520	6.925	0.000	.600	6.925	.300	.520
3	1	6.925	0.000	.600	6.925	.300	.520	7.466	0.000	.600	7.466	.300	.520
4	1	7.466	0.000	.600	7.466	.300	.520	8.008	0.000	.600	8.008	.300	.520
5	1	8.008	0.000	.600	8.008	.300	.520	8.550	0.000	.600	8.550	.300	.520
6	1	8.550	0.000	.600	8.550	.300	.520	9.092	0.000	.600	9.092	.300	.520
7	1	9.092	0.000	.600	9.092	.300	.520	9.634	0.000	.600	9.634	.300	.520
8	1	9.634	0.000	.600	9.634	.300	.520	10.176	0.000	.600	10.176	.300	.520
9	1	10.176	0.000	.600	10.176	.300	.520	10.718	0.000	.600	10.718	.300	.520
10	1	10.718	0.000	.600	10.718	.300	.520	11.260	0.000	.600	11.260	.300	.520
11	1	5.660	.300	.520	5.660	.520	.300	6.503	.300	.520	6.503	.520	.300
12	1	6.503	.300	.520	6.503	.520	.300	6.925	.300	.520	6.925	.520	.300
13	1	6.925	.300	.520	6.925	.520	.300	7.466	.300	.520	7.466	.520	.300
14	1	7.466	.300	.520	7.466	.520	.300	8.008	.300	.520	8.008	.520	.300
15	1	8.008	.300	.520	8.008	.520	.300	8.550	.300	.520	8.550	.520	.300
16	1	8.550	.300	.520	8.550	.520	.300	9.092	.300	.520	9.092	.520	.300
17	1	9.092	.300	.520	9.092	.520	.300	9.634	.300	.520	9.634	.520	.300
18	1	9.634	.300	.520	9.634	.520	.300	10.176	.300	.520	10.176	.520	.300
19	1	10.176	.300	.520	10.176	.520	.300	10.718	.300	.520	10.718	.520	.300
20	1	10.718	.300	.520	10.718	.520	.300	11.260	.300	.520	11.260	.520	.300
21	1	5.660	.520	.300	5.660	.600	-.000	6.503	.520	.300	6.503	.600	-.000
22	1	6.503	.520	.300	6.503	.600	-.000	6.925	.520	.300	6.925	.600	-.000
23	1	6.925	.520	.300	6.925	.600	-.000	7.466	.520	.300	7.466	.600	-.000
24	1	7.466	.520	.300	7.466	.600	-.000	8.008	.520	.300	8.008	.600	-.000
25	1	8.008	.520	.300	8.008	.600	-.000	8.550	.520	.300	8.550	.600	-.000
26	1	8.550	.520	.300	8.550	.600	-.000	9.092	.520	.300	9.092	.600	-.000
27	1	9.092	.520	.300	9.092	.600	-.000	9.634	.520	.300	9.634	.600	-.000
28	1	9.634	.520	.300	9.634	.600	-.000	10.176	.520	.300	10.176	.600	-.000
29	1	10.176	.520	.300	10.176	.600	-.000	10.718	.520	.300	10.718	.600	-.000
30	1	10.718	.520	.300	10.718	.600	-.000	11.260	.520	.300	11.260	.600	-.000
31	1	5.660	.600	-.000	5.660	.520	-.300	6.503	.600	-.000	6.503	.520	-.300
32	1	6.503	.600	-.000	6.503	.520	-.300	6.925	.600	-.000	6.925	.520	-.300
33	1	6.925	.600	-.000	6.925	.520	-.300	7.466	.600	-.000	7.466	.520	-.300
34	1	7.466	.600	-.000	7.466	.520	-.300	8.008	.600	-.000	8.008	.520	-.300
35	1	8.008	.600	-.000	8.008	.520	-.300	8.550	.600	-.000	8.550	.520	-.300
36	1	8.550	.600	-.000	8.550	.520	-.300	9.092	.600	-.000	9.092	.520	-.300
37	1	9.092	.600	-.000	9.092	.520	-.300	9.634	.600	-.000	9.634	.520	-.300
38	1	9.634	.600	-.000	9.634	.520	-.300	10.176	.600	-.000	10.176	.520	-.300
39	1	10.176	.600	-.000	10.176	.520	-.300	10.718	.600	-.000	10.718	.520	-.300
40	1	10.718	.600	-.000	10.718	.520	-.300	11.260	.600	-.000	11.260	.520	-.300
41	1	5.660	.520	-.300	5.660	.300	-.520	6.503	.520	-.300	6.503	.300	-.520
42	1	6.503	.520	-.300	6.503	.300	-.520	6.925	.520	-.300	6.925	.300	-.520
43	1	6.925	.520	-.300	6.925	.300	-.520	7.466	.520	-.300	7.466	.300	-.520
44	1	7.466	.520	-.300	7.466	.300	-.520	8.008	.520	-.300	8.008	.300	-.520
45	1	8.008	.520	-.300	8.008	.300	-.520	8.550	.520	-.300	8.550	.300	-.520
46	1	8.550	.520	-.300	8.550	.300	-.520	9.092	.520	-.300	9.092	.300	-.520
47	1	9.092	.520	-.300	9.092	.300	-.520	9.634	.520	-.300	9.634	.300	-.520
48	1	9.634	.520	-.300	9.634	.300	-.520	10.176	.520	-.300	10.176	.300	-.520
49	1	10.176	.520	-.300	10.176	.300	-.520	10.718	.520	-.300	10.718	.300	-.520
50	1	10.718	.520	-.300	10.718	.300	-.520	11.260	.520	-.300	11.260	.300	-.520
51	1	5.660	.300	-.520	5.660	-.000	-.600	6.503	.300	-.520	6.503	-.000	-.600
52	1	6.503	.300	-.520	6.503	-.000	-.600	6.925	.300	-.520	6.925	-.000	-.600
53	1	6.925	.300	-.520	6.925	-.000	-.600	7.466	.300	-.520	7.466	-.000	-.600
54	1	7.466	.300	-.520	7.466	-.000	-.600	8.008	.300	-.520	8.008	-.000	-.600
55	1	8.008	.300	-.520	8.008	-.000	-.600	8.550	.300	-.520	8.550	-.000	-.600
56	1	8.550	.300	-.520	8.550	-.000	-.600	9.092	.300	-.520	9.092	-.000	-.600
57	1	9.092	.300	-.520	9.092	-.000	-.600	9.634	.300	-.520	9.634	-.000	-.600
58	1	9.634	.300	-.520	9.634	-.000	-.600	10.176	.300	-.520	10.176	-.000	-.600

```
59  1    10.176    .300   -.520  10.176   -.000   -.600  10.718    .300   -.520  10.718   -.000   -.600
60  1    10.718    .300   -.520  10.718   -.000   -.600  11.260    .300   -.520  11.260   -.000   -.600
```

BODY PANEL CENTROID AND CONTROL POINT LOCATIONS EXPRESSED IN THE AERO COORDINATE SYSTEM

PANEL	X_C	Y_C	Z_C	X_{CP}	Y_{CP}	Z_{CP}	AREA	THETA-INCLIN	ALPHA-INCID	CHORD
1	6.082	.150	.560	6.461	.150	.560	.262	-.262	0.000	.843
2	6.714	.150	.560	6.903	.150	.560	.131	-.262	0.000	.422
3	7.195	.150	.560	7.439	.150	.560	.168	-.262	0.000	.542
4	7.737	.150	.560	7.981	.150	.560	.168	-.262	0.000	.542
5	8.279	.150	.560	8.523	.150	.560	.168	-.262	0.000	.542
6	8.821	.150	.560	9.065	.150	.560	.168	-.262	0.000	.542
7	9.363	.150	.560	9.607	.150	.560	.168	-.262	0.000	.542
8	9.905	.150	.560	10.149	.150	.560	.168	-.262	0.000	.542
9	10.447	.150	.560	10.691	.150	.560	.168	-.262	0.000	.542
10	10.989	.150	.560	11.233	.150	.560	.168	-.262	0.000	.542
11	6.082	.410	.410	6.461	.410	.410	.262	-.785	0.000	.843
12	6.714	.410	.410	6.903	.410	.410	.131	-.785	0.000	.422
13	7.195	.410	.410	7.439	.410	.410	.168	-.785	0.000	.542
14	7.737	.410	.410	7.981	.410	.410	.168	-.785	0.000	.542
15	8.279	.410	.410	8.523	.410	.410	.168	-.785	0.000	.542
16	8.821	.410	.410	9.065	.410	.410	.168	-.785	0.000	.542
17	9.363	.410	.410	9.607	.410	.410	.168	-.785	0.000	.542
18	9.905	.410	.410	10.149	.410	.410	.168	-.785	0.000	.542
19	10.447	.410	.410	10.691	.410	.410	.168	-.785	0.000	.542
20	10.989	.410	.410	11.233	.410	.410	.168	-.785	0.000	.542
21	6.082	.560	.150	6.461	.560	.150	.262	-1.309	0.000	.843
22	6.714	.560	.150	6.903	.560	.150	.131	-1.309	0.000	.422
23	7.195	.560	.150	7.439	.560	.150	.168	-1.309	0.000	.542
24	7.737	.560	.150	7.981	.560	.150	.168	-1.309	0.000	.542
25	8.279	.560	.150	8.523	.560	.150	.168	-1.309	0.000	.542
26	8.821	.560	.150	9.065	.560	.150	.168	-1.309	0.000	.542
27	9.363	.560	.150	9.607	.560	.150	.168	-1.309	0.000	.542
28	9.905	.560	.150	10.149	.560	.150	.168	-1.309	0.000	.542
29	10.447	.560	.150	10.691	.560	.150	.168	-1.309	0.000	.542
30	10.989	.560	.150	11.233	.560	.150	.168	-1.309	0.000	.542
31	6.082	.560	-.150	6.461	.560	-.150	.262	-1.833	0.000	.843
32	6.714	.560	-.150	6.903	.560	-.150	.131	-1.833	0.000	.422
33	7.195	.560	-.150	7.439	.560	-.150	.168	-1.833	0.000	.542
34	7.737	.560	-.150	7.981	.560	-.150	.168	-1.833	0.000	.542
35	8.279	.560	-.150	8.523	.560	-.150	.168	-1.833	0.000	.542
36	8.821	.560	-.150	9.065	.560	-.150	.168	-1.833	0.000	.542
37	9.363	.560	-.150	9.607	.560	-.150	.168	-1.833	0.000	.542
38	9.905	.560	-.150	10.149	.560	-.150	.168	-1.833	0.000	.542
39	10.447	.560	-.150	10.691	.560	-.150	.168	-1.833	0.000	.542
40	10.989	.560	-.150	11.233	.560	-.150	.168	-1.833	0.000	.542
41	6.082	.410	-.410	6.461	.410	-.410	.262	-2.356	0.000	.843
42	6.714	.410	-.410	6.903	.410	-.410	.131	-2.356	0.000	.422
43	7.195	.410	-.410	7.439	.410	-.410	.168	-2.356	0.000	.542
44	7.737	.410	-.410	7.981	.410	-.410	.168	-2.356	0.000	.542
45	8.279	.410	-.410	8.523	.410	-.410	.168	-2.356	0.000	.542
46	8.821	.410	-.410	9.065	.410	-.410	.168	-2.356	0.000	.542
47	9.363	.410	-.410	9.607	.410	-.410	.168	-2.356	0.000	.542
48	9.905	.410	-.410	10.149	.410	-.410	.168	-2.356	0.000	.542
49	10.447	.410	-.410	10.691	.410	-.410	.168	-2.356	0.000	.542
50	10.989	.410	-.410	11.233	.410	-.410	.168	-2.356	0.000	.542
51	6.082	.150	-.560	6.461	.150	-.560	.262	-2.880	0.000	.843
52	6.714	.150	-.560	6.903	.150	-.560	.131	-2.880	0.000	.422
53	7.195	.150	-.560	7.439	.150	-.560	.168	-2.880	0.000	.542
54	7.737	.150	-.560	7.981	.150	-.560	.168	-2.880	0.000	.542
55	8.279	.150	-.560	8.523	.150	-.560	.168	-2.880	0.000	.542
56	8.821	.150	-.560	9.065	.150	-.560	.168	-2.880	0.000	.542
57	9.363	.150	-.560	9.607	.150	-.560	.168	-2.880	0.000	.542
58	9.905	.150	-.560	10.149	.150	-.560	.168	-2.880	0.000	.542
59	10.447	.150	-.560	10.691	.150	-.560	.168	-2.880	0.000	.542

```
 60      10.989     .150    -.560   11.233     .150    -.560          .168          -2.880    0.000     .542
```

```
WING PANEL CORNER POINT LOCATIONS EXPRESSED IN THE AERO COORDINATE SYSTEM
1 AND 2 INDICATE WING PANEL LEADING-EDGE POINTS, 3 AND 4 INDICATE TRAILING-EDGE POINTS

 PANEL          X          Y          Z          X          Y          Z          X          Y          Z          X          Y          Z
NO  PARTS        1          1          1          2          2          2          3          3          3          4          4          4


WING NO.  1       YPIVOT =       .600       ZPIVOT =      0.000       WING DIHEDRAL =      0.000 RADIANS

  1   1        6.503       .600     0.000      7.346      1.307     0.000      6.925       .600     0.000      7.683      1.307     0.000
  2   1        6.925       .600     0.000      7.683      1.307     0.000      7.466       .600     0.000      8.117      1.307     0.000
  3   1        7.466       .600     0.000      8.117      1.307     0.000      8.008       .600     0.000      8.550      1.307     0.000
  4   1        8.008       .600     0.000      8.550      1.307     0.000      8.550       .600     0.000      8.984      1.307     0.000
  5   1        8.550       .600     0.000      8.984      1.307     0.000      9.092       .600     0.000      9.417      1.307     0.000
  6   1        9.092       .600     0.000      9.417      1.307     0.000      9.634       .600     0.000      9.851      1.307     0.000
  7   1        9.634       .600     0.000      9.851      1.307     0.000     10.176       .600     0.000     10.284      1.307     0.000
  8   1       10.176       .600     0.000     10.284      1.307     0.000     10.718       .600     0.000     10.718      1.307     0.000
  9   1        7.346      1.307     0.000      8.189      2.015     0.000      7.683      1.307     0.000      8.442      2.015     0.000
 10   1        7.683      1.307     0.000      8.442      2.015     0.000      8.117      1.307     0.000      8.767      2.015     0.000
 11   1        8.117      1.307     0.000      8.767      2.015     0.000      8.550      1.307     0.000      9.092      2.015     0.000
 12   1        8.550      1.307     0.000      9.092      2.015     0.000      8.984      1.307     0.000      9.417      2.015     0.000
 13   1        8.984      1.307     0.000      9.417      2.015     0.000      9.417      1.307     0.000      9.743      2.015     0.000
 14   1        9.417      1.307     0.000      9.743      2.015     0.000      9.851      1.307     0.000     10.068      2.015     0.000
 15   1        9.851      1.307     0.000     10.068      2.015     0.000     10.284      1.307     0.000     10.393      2.015     0.000
 16   1       10.284      1.307     0.000     10.393      2.015     0.000     10.718      1.307     0.000     10.718      2.015     0.000
 17   1        8.189      2.015     0.000      9.032      2.722     0.000      8.442      2.015     0.000      9.201      2.722     0.000
 18   1        8.442      2.015     0.000      9.201      2.722     0.000      8.767      2.015     0.000      9.417      2.722     0.000
 19   1        8.767      2.015     0.000      9.417      2.722     0.000      9.092      2.015     0.000      9.634      2.722     0.000
 20   1        9.092      2.015     0.000      9.634      2.722     0.000      9.417      2.015     0.000      9.851      2.722     0.000
 21   1        9.417      2.015     0.000      9.851      2.722     0.000      9.743      2.015     0.000     10.068      2.722     0.000
 22   1        9.743      2.015     0.000     10.068      2.722     0.000     10.068      2.015     0.000     10.284      2.722     0.000
 23   1       10.068      2.015     0.000     10.284      2.722     0.000     10.393      2.015     0.000     10.501      2.722     0.000
 24   1       10.393      2.015     0.000     10.501      2.722     0.000     10.718      2.015     0.000     10.718      2.722     0.000
 25   1        9.032      2.722     0.000      9.875      3.430     0.000      9.201      2.722     0.000      9.959      3.430     0.000
 26   1        9.201      2.722     0.000      9.959      3.430     0.000      9.417      2.722     0.000     10.068      3.430     0.000
 27   1        9.417      2.722     0.000     10.068      3.430     0.000      9.634      2.722     0.000     10.176      3.430     0.000
 28   1        9.634      2.722     0.000     10.176      3.430     0.000      9.851      2.722     0.000     10.284      3.430     0.000
 29   1        9.851      2.722     0.000     10.284      3.430     0.000     10.068      2.722     0.000     10.393      3.430     0.000
 30   1       10.068      2.722     0.000     10.393      3.430     0.000     10.284      2.722     0.000     10.501      3.430     0.000
 31   1       10.284      2.722     0.000     10.501      3.430     0.000     10.501      2.722     0.000     10.610      3.430     0.000
 32   1       10.501      2.722     0.000     10.610      3.430     0.000     10.718      2.722     0.000     10.718      3.430     0.000
 33   1        9.875      3.430     0.000     10.718      4.137     0.000      9.959      3.430     0.000     10.718      4.137     0.000
 34   1        9.959      3.430     0.000     10.718      4.137     0.000     10.068      3.430     0.000     10.718      4.137     0.000
 35   1       10.068      3.430     0.000     10.718      4.137     0.000     10.176      3.430     0.000     10.718      4.137     0.000
 36   1       10.176      3.430     0.000     10.718      4.137     0.000     10.284      3.430     0.000     10.718      4.137     0.000
 37   1       10.284      3.430     0.000     10.718      4.137     0.000     10.393      3.430     0.000     10.718      4.137     0.000
 38   1       10.393      3.430     0.000     10.718      4.137     0.000     10.501      3.430     0.000     10.718      4.137     0.000
 39   1       10.501      3.430     0.000     10.718      4.137     0.000     10.610      3.430     0.000     10.718      4.137     0.000
 40   1       10.610      3.430     0.000     10.718      4.137     0.000     10.718      3.430     0.000     10.718      4.137     0.000
```

```
          WING PANEL CENTROID AND CONTROL POINT LOCATIONS EXPRESSED IN THE AERO COORDINATE SYSTEM

    PANEL     X         Y         Z         X         Y         Z               AREA          THETA-    ALPHA-      CHORD
               C         C         C         CP        CP        CP                           DIHED     CAMBER

      1      7.099      .941     0.000     7.271      .941     0.000            .268          0.000     0.000       .381
      2      7.535      .941     0.000     7.755      .941     0.000            .345          0.000     0.000       .490
      3      8.024      .941     0.000     8.245      .941     0.000            .345          0.000     0.000       .490
      4      8.514      .941     0.000     8.735      .941     0.000            .345          0.000     0.000       .490
      5      9.004      .941     0.000     9.224      .941     0.000            .345          0.000     0.000       .490
      6      9.494      .941     0.000     9.714      .941     0.000            .345          0.000     0.000       .490
      7      9.983      .941     0.000    10.204      .941     0.000            .345          0.000     0.000       .490
      8     10.473      .941     0.000    10.694      .941     0.000            .345          0.000     0.000       .490
      9      7.896     1.644     0.000     8.030     1.644     0.000            .209          0.000     0.000       .297
     10      8.235     1.644     0.000     8.407     1.644     0.000            .268          0.000     0.000       .382
     11      8.617     1.644     0.000     8.789     1.644     0.000            .268          0.000     0.000       .382
     12      8.999     1.644     0.000     9.171     1.644     0.000            .268          0.000     0.000       .382
     13      9.381     1.644     0.000     9.553     1.644     0.000            .268          0.000     0.000       .382
     14      9.763     1.644     0.000     9.935     1.644     0.000            .268          0.000     0.000       .382
     15     10.145     1.644     0.000    10.317     1.644     0.000            .268          0.000     0.000       .382
     16     10.527     1.644     0.000    10.699     1.644     0.000            .268          0.000     0.000       .382
     17      8.689     2.345     0.000     8.785     2.345     0.000            .149          0.000     0.000       .214
     18      8.933     2.345     0.000     9.057     2.345     0.000            .192          0.000     0.000       .275
     19      9.208     2.345     0.000     9.331     2.345     0.000            .192          0.000     0.000       .275
     20      9.482     2.345     0.000     9.606     2.345     0.000            .192          0.000     0.000       .275
     21      9.757     2.345     0.000     9.881     2.345     0.000            .192          0.000     0.000       .275
     22     10.032     2.345     0.000    10.155     2.345     0.000            .192          0.000     0.000       .275
     23     10.306     2.345     0.000    10.430     2.345     0.000            .192          0.000     0.000       .275
     24     10.581     2.345     0.000    10.704     2.345     0.000            .192          0.000     0.000       .275
     25      9.472     3.037     0.000     9.531     3.037     0.000            .089          0.000     0.000       .131
     26      9.622     3.037     0.000     9.698     3.037     0.000            .115          0.000     0.000       .169
     27      9.791     3.037     0.000     9.867     3.037     0.000            .115          0.000     0.000       .169
     28      9.959     3.037     0.000    10.035     3.037     0.000            .115          0.000     0.000       .169
     29     10.128     3.037     0.000    10.204     3.037     0.000            .115          0.000     0.000       .169
     30     10.297     3.037     0.000    10.372     3.037     0.000            .115          0.000     0.000       .169
     31     10.465     3.037     0.000    10.541     3.037     0.000            .115          0.000     0.000       .169
     32     10.634     3.037     0.000    10.710     3.037     0.000            .115          0.000     0.000       .169
     33     10.184     3.665     0.000    10.209     3.665     0.000            .030          0.000     0.000       .056
     34     10.248     3.665     0.000    10.281     3.665     0.000            .038          0.000     0.000       .072
     35     10.321     3.665     0.000    10.353     3.665     0.000            .038          0.000     0.000       .072
     36     10.393     3.665     0.000    10.425     3.665     0.000            .038          0.000     0.000       .072
     37     10.465     3.665     0.000    10.498     3.665     0.000            .038          0.000     0.000       .072
     38     10.537     3.665     0.000    10.570     3.665     0.000            .038          0.000     0.000       .072
     39     10.610     3.665     0.000    10.642     3.665     0.000            .038          0.000     0.000       .072
     40     10.682     3.665     0.000    10.714     3.665     0.000            .038          0.000     0.000       .072

DESCRIPTION OF CASE REQUESTED

SYMMETRICAL CONFIGURATION - PANELS LOCATED ON BOTH SIDES OF X-Z PLANE(SYM = 1.)

CASE = 2.           CALCULATE CL, GIVEN SHAPE

CPCALC = 1.         NON-LINEAR CP

POLAR = 1.          POLARS REQUESTED

THICK = 1.          WING THICKNESS PRESSURES TO BE ADDED

VOUT = 0.           VELOCITY COMPONENTS NOT TO BE PRINTED

MACH NUMBER =     2.0000
```

```
POINT ABOUT WHICH THE MOMENTS ARE TO BE COMPUTED
 X-COORDINATE =    0.0000
 Z-COORDINATE =    0.0000

REFERENCE CHORD LENGTH =    1.0000

WING REFERENCE AREA =    7.4542

WING SEMI-SPAN =     1.0000

HEIGHT OF WING PLANE ABOVE BODY AXIS =    0.0000

INCLINATION OF BODY AXIS WITH RESPECT TO DEFINING AXIS =    0.0000  DEG.

ANGLE OF ATTACK WITH RESPECT TO BODY AXIS =    5.0000  DEG.

BODY YAW =       5.00 DEG

BODY ROLL =       0.00 DEG
```

```
PRESSURES, FORCES, AND MOMENTS ON ISOLATED BODY

CD =    .0194    CL =    .0311    CB =   -.0313    CPM =   -.0398    CRM =   0.0000    CYM =    .0386

BODY PRESSURE COEFFICIENTS(CP)

THETA(DEG.)     0.00     30.00     60.00     90.00    120.00    150.00    180.00    210.00    240.00    270.00    300.00    330.0
      X
   0.0000      .0303     .0460     .0917     .1619     .2188     .2190     .1625     .0922     .0462     .0303     .0298     .0298
    .3000      .0299     .0456     .0914     .1615     .2183     .2185     .1620     .0918     .0458     .0299     .0294     .0294
    .6000      .0300     .0454     .0909     .1608     .2176     .2178     .1613     .0913     .0456     .0301     .0297     .0297
    .9000      .0325     .0482     .0943     .1649     .2221     .2224     .1654     .0947     .0485     .0325     .0320     .0320
   1.2000      .0323     .0486     .0951     .1661     .2235     .2237     .1666     .0956     .0488     .0324     .0315     .0315
   1.5000      .0283     .0445     .0906     .1608     .2175     .2178     .1613     .0910     .0447     .0283     .0273     .0274
   1.8000      .0248     .0403     .0854     .1546     .2106     .2108     .1551     .0858     .0405     .0248     .0242     .0243
   2.1000      .0281     .0431     .0879     .1572     .2135     .2138     .1577     .0884     .0433     .0282     .0281     .0281
   2.4000      .0533     .0698     .1186     .1935     .2541     .2544     .1941     .1191     .0701     .0534     .0530     .0530
   2.7000      .0124     .0288     .0735     .1410     .1956     .1958     .1415     .0738     .0289     .0125     .0110     .0110
   3.0000     -.1156    -.1009    -.0671    -.0186     .0204     .0205    -.0184    -.0671    -.1010    -.1158    -.1188    -.1187
   3.8571     -.0687    -.0674    -.0425     .0030     .0419     .0420     .0032    -.0424    -.0675    -.0689    -.0613    -.0612
   4.7143     -.0518    -.0692    -.0597    -.0242     .0100     .0101    -.0240    -.0596    -.0693    -.0520    -.0302    -.0301
   5.5714     -.0342    -.0608    -.0591    -.0292     .0018     .0019    -.0290    -.0590    -.0609    -.0344    -.0056    -.0055
   6.4286     -.0269    -.0570    -.0589    -.0324    -.0036    -.0035    -.0321    -.0588    -.0571    -.0271     .0039     .0041
   7.2857     -.0215    -.0519    -.0548    -.0297    -.0023    -.0021    -.0295    -.0547    -.0520    -.0217     .0094     .0095
   8.1429     -.0188    -.0483    -.0510    -.0264     .0005     .0006    -.0262    -.0509    -.0484    -.0191     .0111     .0112
   9.0000     -.0171    -.0453    -.0472    -.0225     .0043     .0044    -.0223    -.0471    -.0454    -.0173     .0118     .0119
   9.8571     -.0168    -.0438    -.0449    -.0199     .0070     .0071    -.0196    -.0448    -.0439    -.0170     .0111     .0112
  10.7143     -.0182    -.0443    -.0448    -.0195     .0073     .0074    -.0193    -.0447    -.0444    -.0184     .0089     .0090
  11.5714     -.0210    -.0468    -.0471    -.0219     .0048     .0049    -.0217    -.0470    -.0469    -.0212     .0059     .0060
  12.4286     -.0246    -.0508    -.0515    -.0269    -.0007    -.0006    -.0267    -.0515    -.0509    -.0248     .0026     .0027
  13.2857     -.0287    -.0560    -.0579    -.0344    -.0089    -.0088    -.0342    -.0578    -.0561    -.0289    -.0006    -.0004
  14.1429     -.0353    -.0645    -.0681    -.0463    -.0222    -.0221    -.0462    -.0681    -.0646    -.0356    -.0058    -.0057
  15.0000      .0087    -.0267    -.0344    -.0134     .0115     .0116    -.0132    -.0343    -.0268     .0085     .0431     .0432
```

```
INCREMENTAL PRESSURES, FORCES, AND MOMENTS ON BODY PANELS DUE TO WING

CD =    .0006    CL =    .0265    CB =    .0197    CPM =    .1628    CRM =   0.0000    CYM =    .2180

BODY PANEL PRESSURE COEFFICIENT(CP)

THETA(DEG.)     15.00     45.00     75.00    105.00    135.00    165.00    195.00    225.00    255.00    285.00    315.00    345.0

  ROW NO.
    1          -.0000    -.0000     .0000    -.0000     .0000    -.0000    -.0000     .0000    -.0000     .0000    -.0000    -.0000
    2           .0001     .0013    -.0521     .0747     .0001     .0000     .0000     .0001     .0747    -.0521     .0013     .0001
    3           .0021    -.0209    -.0991     .1335     .0386     .0007     .0007     .0386     .1335    -.0991    -.0209     .0021
    4          -.0064    -.0566    -.0748     .0831     .0853     .0204     .0204     .0853     .0831    -.0748    -.0566    -.0064
    5          -.0390    -.0486    -.0695     .0573     .0773     .0548     .0548     .0773     .0573    -.0695    -.0486    -.0390
    6          -.0504    -.0424    -.0675     .0352     .0706     .0735     .0735     .0706     .0352    -.0675    -.0424    -.0504
    7          -.0497    -.0411    -.0731     .0194     .0698     .0760     .0760     .0698     .0194    -.0731    -.0411    -.0497
    8          -.0454    -.0410    -.0840     .0033     .0702     .0738     .0738     .0702     .0033    -.0840    -.0410    -.0454
    9          -.0427    -.0399    -.0977    -.0110     .0694     .0736     .0736     .0694    -.0110    -.0977    -.0399    -.0427
   10          -.0408    -.0370    -.0913    -.0635     .0652     .0731     .0731     .0652    -.0635    -.0913    -.0370    -.0408


BODY PANEL SLOPE(DR/DX)

THETA(DEG.)     15.00     45.00     75.00    105.00    135.00    165.00    195.00    225.00    255.00    285.00    315.00    345.0

  ROW NO.
    1          -.0618    -.0003     .0613     .1066     .1233     .1068     .0616     .0001    -.0615    -.1068    -.1235    -.1070
    2          -.0621    -.0006     .0610     .1063     .1230     .1065     .0614    -.0001    -.0617    -.1070    -.1237    -.1072
    3          -.0622    -.0007     .0609     .1062     .1229     .1064     .0613    -.0007    -.0618    -.1071    -.1238    -.1073
    4          -.0621    -.0006     .0610     .1063     .1230     .1065     .0614    -.0002    -.0617    -.1070    -.1237    -.1072
    5          -.0618    -.0003     .0613     .1066     .1233     .1066     .0617     .0001    -.0615    -.1067    -.1234    -.1070
    6          -.0614     .0002     .0618     .1070     .1237     .1072     .0621     .0006    -.0610    -.1063    -.1230    -.1065
    7          -.0608     .0008     .0624     .1076     .1243     .1078     .0627     .0012    -.0604    -.1057    -.1224    -.1059
    8          -.0606     .0010     .0626     .1078     .1245     .1081     .0629     .0014    -.0602    -.1055    -.1222    -.1057
    9          -.0610     .0005     .0621     .1074     .1241     .1076     .0625     .0010    -.0606    -.1059    -.1226    -.1061
   10          -.0621    -.0006     .0610     .1063     .1230     .1065     .0613    -.0002    -.0618    -.1071    -.1238    -.1073
```

```
PRESSURFS, FORCES, AND MGMENTS ON WING PANELS IN PRESENLE OF BODY

CD =    .0479     CL =    .2515     CB =    .0023     CPM =  -2.3062     CRM =    .4876     CYM =   -.0704

SECTION CD DISTRIBUTION

 SPANWISE STATION        1          2          3          4          5
                      .04004     .04598     .05590     .06010     .05655

SECTION CL DISTRIBUTION

 SPANWISE STATION        1          2          3          4          5
                      .20650     .24106     .29594     .31941     .30264

SECTION CB DISTRIBUTION

 SPANWISE STATION        1          2          3          4          5
                      .00191     .00216     .00261     .00279     .00261

UPPER SURFACE WING PANEL PRESSURE COEFFICIENTS(CP)

 SPANWISE STATION        1          2          3          4          5
 CHORDWISE STATION
        1             -.16365    -.13427    -.11961    -.11893    -.11696
        2             -.11344    -.13776    -.12237    -.11716    -.11716
        3             -.07038    -.11456    -.12689    -.11576    -.11729
        4             -.04761    -.08883    -.12671    -.11619    -.11727
        5             -.03605    -.06455    -.11590    -.11779    -.11710
        6             -.03269    -.05036    -.10098    -.11968    -.11679
        7             -.03245    -.04152    -.08329    -.12145    -.11637
        8             -.03477    -.03960    -.06671    -.12199    -.11588

LOWER SURFACE WING PANEL PRESSURE COEFFECIENTS(CP)

 SPANWISE STATION        1          2          3          4          5
 CHORDWISE STATION
        1              .27185     .23140     .21016     .19453     .18950
        2              .20897     .23461     .21571     .19850     .18910
        3              .15329     .20306     .22058     .20354     .18933
        4              .12559     .17036     .21891     .20680     .18999
        5              .11323     .13837     .20232     .20988     .19094
        6              .11096     .12083     .18607     .21216     .19218
        7              .11046     .11093     .16210     .21332     .19367
        8              .11115     .11066     .13960     .21221     .19546

WING PANEL PRESSURE DIFFERENCE(CL)

 SPANWISE STATION        1          2          3          4          5
 CHORDWISE STATION
        1              .41652     .35227     .31724     .30407     .29760
        2              .30228     .35879     .32503     .30500     .29755
        3              .20839     .30408     .33450     .30728     .29789
        4              .16033     .24716     .33269     .31038     .29840
        5              .13721     .19346     .30627     .31477     .29895
        6              .13106     .16311     .27440     .31891     .29956
        7              .12963     .14514     .23457     .32201     .30023
        8              .13159     .14278     .19736     .32184     .30104
```

```
UPPER SURFACE WING PANEL SLOPE(DZ/DX)

 SPANWISE STATION          1         2         3         4         5
 CHORDWISE STATION
        1              -.04367   -.04367   -.04367   -.04367   -.04367
        2              -.04367   -.04367   -.04367   -.04367   -.04367
        3              -.04367   -.04367   -.04367   -.04367   -.04367
        4              -.04367   -.04367   -.04367   -.04367   -.04367
        5              -.04367   -.04367   -.04367   -.04367   -.04367
        6              -.04367   -.04367   -.04367   -.04367   -.04367
        7              -.04367   -.04367   -.04367   -.04367   -.04367
        8              -.04367   -.04367   -.04367   -.04367   -.04367

LOWER SURFACE WING PANEL SLOPE(DZ/DX)

 SPANWISE STATION          1         2         3         4         5
 CHORDWISE STATION
        1              -.13087   -.13087   -.13087   -.13087   -.13087
        2              -.13087   -.13087   -.13087   -.13087   -.13087
        3              -.13087   -.13087   -.13087   -.13087   -.13087
        4              -.13087   -.13087   -.13087   -.13087   -.13087
        5              -.13087   -.13087   -.13087   -.13087   -.13087
        6              -.13087   -.13087   -.13087   -.13087   -.13087
        7              -.13087   -.13087   -.13087   -.13087   -.13087
        8              -.13087   -.13087   -.13087   -.13087   -.13087

WING CAMBER SLOPE(DZ/DX)

 SPANWISE STATION          1         2         3         4         5
 CHORDWISE STATION
        1              -.08727   -.08727   -.08727   -.08727   -.08727
        2              -.08727   -.08727   -.08727   -.08727   -.08727
        3              -.08727   -.08727   -.08727   -.08727   -.08727
        4              -.08727   -.08727   -.08727   -.08727   -.08727
        5              -.08727   -.08727   -.08727   -.08727   -.08727
        6              -.08727   -.08727   -.08727   -.08727   -.08727
        7              -.08727   -.08727   -.08727   -.08727   -.08727
        8              -.08727   -.08727   -.08727   -.08727   -.08727

WING CHORD LENGTHS(C)
                       3.80911   2.97057   2.13560   1.31133    .56200
```

```
FORCES AND MOMENTS ON WING-BODY COMBINATION

CD =    .0679     CL =    .3090     CB =   0.0000     CPM =  -2.1831     CRM =    .4876     CYM =    .1861

BODY PRESSURE COEFFICIENTS (CP), INCLUDING THE EFFECT OF WING

   THETA (DEG)        0.0000   30.0000   60.0000   90.0000  120.0000  150.0000  180.0000  210.0000  240.0000  270.0000
                    300.0000  330.0000
       X
     0.0000           .03026    .04598    .09175    .16195    .21881    .21904    .16246    .09219    .04619    .03030
                     .02975    .02976
      .3000           .02989    .04563    .09137    .16150    .21829    .21853    .16200    .09181    .04584    .02993
                     .02935    .02937
      .6000           .03002    .04543    .09089    .16083    .21756    .21779    .16134    .09133    .04564    .03005
                     .02971    .02972
      .9000           .03247    .04825    .09429    .16493    .22214    .22238    .16545    .09474    .04847    .03251
                     .03198    .03199
     1.2000           .03232    .04860    .09511    .16609    .22347    .22371    .16661    .09557    .04882    .03236
                     .03150    .03151
     1.5000           .02825    .04447    .09055    .16078    .21755    .21778    .16128    .09099    .04468    .02828
                     .02734    .02735
     1.8000           .02480    .04033    .08542    .15456    .21059    .21081    .15505    .08584    .04052    .02482
                     .02425    .02426
     2.1000           .02814    .04308    .08792    .15722    .21352    .21375    .15772    .08836    .04329    .02817
                     .02809    .02810
     2.4000           .05331    .06980    .11857    .19349    .25410    .25437    .19408    .11911    .07008    .05338
                     .05302    .05302
     2.7000           .01245    .02875    .07346    .14105    .19559    .19580    .14150    .07385    .02892    .01246
                     .01100    .01102
     3.0000          -.11560   -.10090   -.06711   -.01855    .02037    .02045   -.01840   -.06706   -.10099   -.11575
                    -.11881   -.11874
     3.8571          -.06874   -.06737   -.04246    .00299    .04191    .04201    .00318   -.04239   -.06747   -.06892
                    -.06127   -.06118
     4.7143          -.05179   -.06921   -.05971   -.02420    .00997    .01008   -.02400   -.05962   -.06931   -.05199
                    -.03016   -.03006
     5.5714          -.03416   -.06079   -.05911   -.02921    .00183    .00195   -.02899   -.05902   -.06089   -.03437
                    -.00563   -.00552
     6.4286          -.02691   -.05698   -.05888   -.03235   -.00361   -.00350   -.03214   -.05878   -.05708   -.02713
                     .00394    .00406
     7.2857          -.02068   -.13728   -.01057   -.03460   -.00521    .00160   -.01781   -.03434   -.02272    .01673
                     .05708    .06657
     8.1429          -.03080   -.13032    .03904   -.01613    .00859    .01378   -.00621   -.02315   -.01247    .02544
                     .06436    .07340
     9.0000          -.07069   -.11447    .02359    .03360    .03826    .03388    .00976   -.00963   -.00119    .03438
                     .07138    .07977
     9.8571          -.07248   -.11631    .02155    .04803    .05959    .05270    .02542    .00335    .00929    .04240
                     .07747    .08511
    10.7143          -.06885   -.13201    .01945    .04775    .06858    .06470    .03692    .01333    .01740    .04857
                     .08207    .08901
    11.5714          -.02101   -.04678   -.04710   -.02192    .00479    .00490   -.02172   -.04701   -.04688   -.02122
                     .00590    .00601
    12.4286          -.02461   -.05078   -.05154   -.02692   -.00067   -.00057   -.02673   -.05146   -.05089   -.02483
                     .00260    .00271
    13.2857          -.02867   -.05601   -.05788   -.03436   -.00891   -.00881   -.03418   -.05781   -.05613   -.02890
                    -.00056   -.00044
    14.1429          -.03532   -.06449   -.06810   -.04636   -.02221   -.02212   -.04618   -.06806   -.06464   -.03557
                    -.00579   -.00567
    15.0000           .00873   -.02671   -.03442   -.01343    .01151    .01162   -.01321   -.03431   -.02681    .00851
                     .04306    .04318
```

```
AERODYNAMIC PRESSURE ON THE BODY - - CENTROID OF FINITE ELEMENT AS GENERATED BY NSWC CODE IN THE
  SAME LOCAL COORDINATE SYSTEM

ELEMENT      X          THETA        PRESSURE         ELEMENT      X          THETA        PRESSURE

    1    .269E+00     .345E+03     .145E+01              2    .269E+00     .315E+03     .264E+01
    3    .269E+00     .285E+03     .516E+01              4    .269E+00     .255E+03     .796E+01
    5    .269E+00     .225E+03     .937E+01              6    .269E+00     .195E+03     .797E+01
    7    .269E+00     .165E+03     .797E+01              8    .269E+00     .135E+03     .937E+01
    9    .269E+00     .105E+03     .796E+01             10    .269E+00     .750E+02     .516E+01
   11    .269E+00     .450E+02     .264E+01
```

```
TOTAL LIFTING PRESSURE ON THE WING AT THE CONTROL POINT

      X          Y          PRESSURE

    7.2708      .9406      .1780E+02
    7.7551      .9406      .1327E+02
    8.2448      .9406      .9206E+01
    8.7345      .9406      .7129E+01
    9.2243      .9406      .6144E+01
    9.7140      .9406      .5913E+01
   10.2038      .9406      .5882E+01
   10.6935      .9406      .6006E+01
    8.0296     1.6443      .1505E+02
    8.4073     1.6443      .1533E+02
    8.7893     1.6443      .1307E+02
    9.1712     1.6443      .1067E+02
    9.5531     1.6443      .8352E+01
    9.9350     1.6443      .7046E+01
   10.3170     1.6443      .6275E+01
   10.6989     1.6443      .6185E+01
    8.7853     2.3449      .1357E+02
    9.0568     2.3449      .1392E+02
    9.3316     2.3449      .1430E+02
    9.6060     2.3449      .1423E+02
    9.8805     2.3449      .1310E+02
   10.1551     2.3449      .1182E+02
   10.4297     2.3449      .1010E+02
   10.7043     2.3449      .8492E+01
    9.5312     3.0366      .1290E+02
    9.6980     3.0366      .1299E+02
    9.8666     3.0366      .1314E+02
   10.0352     3.0366      .1329E+02
   10.2038     3.0366      .1349E+02
   10.3724     3.0366      .1366E+02
   10.5410     3.0366      .1378E+02
   10.7096     3.0366      .1376E+02
   10.2094     3.6654      .1261E+02
   10.2808     3.6654      .1261E+02
   10.3531     3.6654      .1262E+02
   10.4254     3.6654      .1265E+02
   10.4976     3.6654      .1268E+02
   10.5699     3.6654      .1272E+02
   10.6421     3.6654      .1276E+02
   10.7144     3.6654      .1281E+02
```

```
AERODYNAMIC PRESSURE ON THE WING - - CENTROID OF FINITE ELEMENT AS GENERATED BY NSWC CODE IN THE
  SAME LOCAL COORDINATE SYSTEM

     TOTAL NUMBER OF WINGS      1

     WING  NO.     1

ELEMENT        X            Y            PRESSURE        ELEMENT        X            Y            PRESSURE


  400     .697E+01     .797E+00     .194E+02        401     .756E+01     .895E+00     .150E+02
  402     .826E+01     .895E+00     .884E+01        403     .896E+01     .895E+00     .633E+01
  404     .966E+01     .895E+00     .583E+01        405     .104E+02     .895E+00     .591E+01
  406     .767E+01     .139E+01     .156E+02        407     .826E+01     .148E+01     .142E+02
  408     .896E+01     .148E+01     .108E+02        409     .966E+01     .148E+01     .706E+01
  410     .104E+02     .148E+01     .592E+01


.........END OF COMPUTATIONS .......
```

5.4.2 High-Supersonic Flow

EXPERIMENT CASE FOR CHECKING GENERALIZED MISSILE ATTITUDE MODE

```
SUNIFID
ICASE  = 3,
NTRANS = 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,   0, 0, 0,
IPUNCH = 0,
IRW    = 0,
POLAR  = 0.0,
ISOLID = 1,
SID    = 0.1E+01, 0.1E+01, 0.1E+01, 0.1E+01, 0.1E+01, 0.1E+01, 0.1E+01, 0.1E+01, 0.1E+01, 0.1E+01,
         0.1E+01, 0.1E+01, 0.1E+01, 0.1E+01, 0.1E+01, 0.1E+01, 0.1E+01, 0.1E+01, 0.1E+01, 0.1E+01,
NBPFL  = 15,
XB     = 0.0, 0.5E+00, 0.1E+01, 0.15E+01, 0.2E+01, 0.25E+01, 0.3E+01,   0.33E+01, 0.3403E+01, 0.35E+01,
         0.55E+01, 0.75E+01, 0.95E+01, 0.12E+02, 0.15E+02, 0.0, 0.0,    0.0, 0.0, 0.0, 0.0, 0.0,
RB     = 0.0, 0.881E-01, 0.1763E+00, 0.2664E+00, 0.3526E+00, 0.4408E+00, 0.5289E+00, 0.5818E+00, 0.6E+00,
         0.6E+00, 0.6E+00, 0.6E+00, 0.6E+00, 0.6E+00, 0.0, 0.0, 0.0,    0.0, 0.0, 0.0, 0.0, 0.0,
ZB     = 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,      0.0, 0.0, 0.0, 0.0, 0.0,
         0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,      0.0, 0.0, 0.0, 0.0, 0.0,
RP     = 0.0,
NWPI   = 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,   0, 0, 0,
YAW    = 0.0,
BROLL  = 0.0,
NWING  = 1,
IFORM  = 2, 0, 0, 0, 0, 0, 0, 0, 0, 0,
PIVOT  = 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
THICK  = 0.1E+01,
DIHED  = 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
XWI    = 0.6503E+01, 0.10718E+02, 0.10718E+02, 0.10718E+02, 0.0, 0.0,   0.0, 0.0, 0.0, 0.0, 0.0,
         0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,      0.0, 0.0, 0.0, 0.0, 0.0,
YWI    = 0.6E+00, 0.6E+00, 0.4137E+01, 0.4137E+01, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
         0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,      0.0, 0.0, 0.0, 0.0, 0.0,
ZWI    = 0.0, 0.184E+00, 0.0, 0.184E+00, 0.0, 0.0, 0.0, 0.0, 0.0,    0.0, 0.0, 0.0, 0.0, 0.0,
         0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,      0.0, 0.0, 0.0, 0.0, 0.0,
XNACEL = 0.0,
NACELP = 0,
XN     = 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,      0.0, 0.0, 0.0, 0.0, 0.0,
         0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,      0.0, 0.0, 0.0, 0.0, 0.0,
RN     = 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,      0.0, 0.0, 0.0, 0.0, 0.0,
         0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,      0.0, 0.0, 0.0, 0.0, 0.0,
ZN     = 0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,      0.0,  0.0,  0.0,  0.0,  0.0,
         0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,      0.0,  0.0,  0.0,  0.0,  0.0,

NWPANL = 1,

XLG    = 0.6503E+01,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

XTG    = 0.10718E+02,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

XMACH  = 0.258E+01,

PINF   = 0.147E+02,

DADEG  = 0.0,

ARB    = 0.5E+01,

ARW    = 0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,  0.0,

ARN    = 0.0,  0.0,

$END
```

```
PRESSURES, FORCES AND MOMENTS ON ISOLATED BODY

CD =    .0212    CL =    .0987    CB =    .0000    CPM =    .3613    CRM =   0.0000    CYM =   0.0000

 BODY PRESSURE COEFFICIENTS (CP)

THETA(DEG)     0.0000   30.0000   60.0000   90.0000  120.0000  150.0000  180.0000

    X
  0.0000     .0362     .0379     .0465     .0675     .0991     .1291     .1415
   .3125     .0391     .0408     .0495     .0708     .1029     .1332     .1457
   .9375     .0410     .0427     .0515     .0729     .1052     .1357     .1484
  1.5625     .0396     .0413     .0500     .0713     .1035     .1338     .1464
  2.1875     .0344     .0362     .0448     .0657     .0973     .1272     .1395
  2.8125     .0327     .0344     .0431     .0640     .0954     .1251     .1374
  3.4375    -.0794    -.0782    -.0729    -.0605    -.0422    -.0250    -.0179
  4.0625    -.0685    -.0672    -.0614    -.0480    -.0281    -.0094    -.0016
  4.6875    -.0570    -.0556    -.0493    -.0347    -.0132     .0070     .0153
  5.3125    -.0475    -.0460    -.0392    -.0237    -.0009     .0205     .0294
  5.9375    -.0395    -.0379    -.0309    -.0146     .0093     .0318     .0410
  6.5625    -.0322    -.0305    -.0232    -.0062     .0187     .0421     .0517
  7.1875    -.0254    -.0236    -.0160     .0016     .0274     .0516     .0616
  7.8125    -.0190    -.0172    -.0093     .0088     .0355     .0605     .0708
  8.4375    -.0131    -.0112    -.0031     .0155     .0430     .0687     .0793
  9.0625    -.0075    -.0056     .0027     .0218     .0499     .0762     .0871
  9.6875    -.0024    -.0005     .0080     .0275     .0563     .0832     .0943
 10.3125     .0017     .0036     .0123     .0322     .0614     .0887     .1000
 10.9375     .0047     .0067     .0155     .0356     .0652     .0929     .1043
 11.5625     .0068     .0087     .0176     .0379     .0677     .0956     .1071
 12.1875     .0076     .0096     .0185     .0389     .0688     .0967     .1083
 12.8125     .0078     .0098     .0187     .0391     .0690     .0969     .1085
 13.4375     .0073     .0093     .0182     .0385     .0683     .0962     .1077
 14.0625     .0059     .0079     .0167     .0369     .0666     .0944     .1058
```

```
FORCES AND MOMENTS ON WING

CD =    .0191    CL =    .1677    CB =    .0000    CPM =  -1.5614    CRM =    .2992    CYM =   -.0339

 UPPER SURFACE WING PANEL PRESSURE COEFF. (CP)

 SPANWISE STATION         1          2          3          4          5
 CHORDWISE STATION
       1              -.03429    -.03429    -.03429    -.03429    -.03429
       2              -.03429    -.03429    -.03429    -.03429    -.03429
       3              -.03429    -.03429    -.03429    -.03429    -.03429
       4              -.03429    -.03429    -.03429    -.03429    -.03429
       5              -.03429    -.03429    -.03429    -.03429    -.03429
       6              -.03429    -.03429    -.03429    -.03429    -.03429
       7              -.03429    -.03429    -.03429    -.03429    -.03429
       8              -.03429    -.03429    -.03429    -.03429    -.03429

 LOWER SURFACE WING PANEL PRESSURE COEFF. (CP)

 SPANWISE STATION         1          2          3          4          5
 CHORDWISE STATION
       1               .13455     .13455     .13455     .13455     .13455
       2               .13455     .13455     .13455     .13455     .13455
       3               .13455     .13455     .13455     .13455     .13455
       4               .13455     .13455     .13455     .13455     .13455
       5               .13455     .13455     .13455     .13455     .13455
       6               .13455     .13455     .13455     .13455     .13455
       7               .13455     .13455     .13455     .13455     .13455
       8               .13455     .13455     .13455     .13455     .13455
```

```
AERODYNAMIC PRESSURE ON THE BODY - - CENTROID OF FINITE ELEMENT AS GENERATED BY NSWC CODE IN THE
  SAME LOCAL COOPDINATE SYSTEM

ELEMENT     X          THETA       PRESSURE          ELEMENT     X          THETA       PRESSURE

    1   .269E+00    .345E+03    .269E+01              2    .269E+00    .315E+03    .296E+01
    3   .269E+00    .285E+03    .403E+01              4    .269E+00    .255E+03    .589E+01
    5   .269E+00    .225E+03    .816E+01              6    .269E+00    .195E+03    .962E+01
    7   .269E+00    .165E+03    .962E+01              8    .269E+00    .135E+03    .816E+01
    9   .269E+00    .105E+03    .589E+01             10    .269E+00    .750E+02    .403E+01
   11   .269E+00    .450E+02    .296E+01
```

```
FORCES AND MOMENTS ON WING

CD =    .0191    CL =    .1677    CB =    .0000    CPM =  -1.5614    CRM =    .2992    CYM =  -.0339

 UPPER SURFACE WING PANEL PRESSURE COEFF. (CP)

 SPANWISE STATION         1         2         3         4         5
 CHORDWISE STATION
        1             -.03429   -.03429   -.03429   -.03429   -.03429
        2             -.03429   -.03429   -.03429   -.03429   -.03429
        3             -.03429   -.03429   -.03429   -.03429   -.03429
        4             -.03429   -.03429   -.03429   -.03429   -.03429
        5             -.03429   -.03429   -.03429   -.03429   -.03429
        6             -.03429   -.03429   -.03429   -.03429   -.03429
        7             -.03429   -.03429   -.03429   -.03429   -.03429
        8             -.03429   -.03429   -.03429   -.03429   -.03429

 LOWER SURFACE WING PANEL PRESSURE COEFF. (CP)

 SPANWISE STATION         1         2         3         4         5
 CHORDWISE STATION
        1              .13455    .13455    .13455    .13455    .13455
        2              .13455    .13455    .13455    .13455    .13455
        3              .13455    .13455    .13455    .13455    .13455
        4              .13455    .13455    .13455    .13455    .13455
        5              .13455    .13455    .13455    .13455    .13455
        6              .13455    .13455    .13455    .13455    .13455
        7              .13455    .13455    .13455    .13455    .13455
        8              .13455    .13455    .13455    .13455    .13455
```

```
TOTAL LIFTING PRESSURE ON THE WING AT THE CONTROL POINT

       X          Y          PRESSURE

    7.2708      .9406      .1156E+02
    7.7551      .9406      .1156E+02
    8.2448      .9406      .1156E+02
    8.7345      .9406      .1156E+02
    9.2243      .9406      .1156E+02
    9.7140      .9406      .1156E+02
   10.2038      .9406      .1156E+02
   10.6935      .9406      .1156E+02
    8.0296     1.6443      .1156E+02
    8.4073     1.6443      .1156E+02
    8.7893     1.6443      .1156E+02
    9.1712     1.6443      .1156E+02
    9.5531     1.6443      .1156E+02
    9.9350     1.6443      .1156E+02
   10.3170     1.6443      .1156E+02
   10.6989     1.6443      .1156E+02
    8.7853     2.3449      .1156E+02
    9.0568     2.3449      .1156E+02
    9.3314     2.3449      .1156E+02
    9.6060     2.3449      .1156E+02
    9.8805     2.3449      .1156E+02
   10.1551     2.3449      .1156E+02
   10.4297     2.3449      .1156E+02
   10.7043     2.3449      .1156E+02
    9.5312     3.0366      .1156E+02
    9.6980     3.0366      .1156E+02
    9.8666     3.0366      .1156E+02
   10.0352     3.0365      .1156E+02
   10.2038     3.0366      .1156E+02
   10.3724     3.0366      .1156E+02
   10.5410     3.0366      .1156E+02
   10.7096     3.0366      .1156E+02
   10.2094     3.6654      .1156E+02
   10.2808     3.6654      .1156E+02
   10.3531     3.6654      .1156E+02
   10.4254     3.6654      .1156E+02
   10.4976     3.6654      .1156E+02
   10.5699     3.6654      .1156E+02
   10.6421     3.6654      .1156E+02
   10.7144     3.6654      .1156E+02
```

```
AERODYNAMIC PRESSURE ON THE WING - - CENTROID OF FINITE ELEMENT AS GENERATED BY NSWC CODE IN THE
  SAME LOCAL COORDINATE SYSTEM

      TOTAL NUMBER OF WINGS      1

      WING  NO.     1

ELEMENT        X          Y          PRESSURE         ELEMENT        X          Y          PRESSURE


  400     .697F+01    .797E+00     .116E+02           401     .756E+01    .895E+00     .116E+02
  402     .826E+01    .895E+00     .116E+02           403     .896E+01    .895E+00     .116E+02
  404     .966E+01    .895E+00     .116E+02           405     .104E+02    .895E+00     .116E+02
  406     .767E+01    .139E+01     .116E+02           407     .826E+01    .148E+01     .116E+02
  408     .896E+01    .148E+01     .116E+02           409     .966E+01    .148E+01     .116E+02
  410     .104E+02    .148E+01     .116E+02


.........END OF COMPUTATIONS .......
```

Section 6
CONCLUSIONS AND RECOMMENDATIONS

The MDAC aerodynamic code has been extended to treat (a) high supersonic speeds ($M_\infty > 2.5$) and (b) generalized missile attitudes. As a result it is capable of analyzing the missile under more realistic flight conditions. Effort for input preparation remains the same for all practical purposes because only a few inputs were added for the specification of attitudes. Computing time for high supersonic flow analysis will be only a small fraction of that for the low-speed case. The original flexibility in the coordinate system has been maintained for the convenience of users. Outputs are produced as computer printout and punch cards (if desired) in the format of NASTRAN bulk data.

With these additional capabilities incorporated, the code represents a versatile tool for aerodynamic analysis of missiles and forms a unique link between the flight dynamics and structural analysis of the missile under a wide variety of flight conditions that it may encounter in actual operation. Furthermore, the large effort necessary for preparing the input to the NASTRAN is reduced when it is used in combination with the NSWC computer codes PING or BING.

For missile flight at high supersonic speeds ($2 \leq M_\infty \leq 8$) the problem of aerodynamic heating becomes important and needs to be analyzed. As an example, for an insulated flat plate aligned parallel to the flow, the surface temperature can reach anywhere from 1000°R to 7000°R for a speed range of $M_\infty = 2$ to 8. Additionally, during maneuvering, severe heating may occur as a result of shock impingement on the missile, e.g., nose bow shock-wing interaction. Thus, the induced thermal stress can seriously affect the structural integrity of the missile, and failure of a missile component can result. Consequently, development of capability for analyzing the aerodynamic heating to complement the pressure-load calculation will be an appropriate task and is recommended as a future effort. The required flow-field quantities such as the pressure, velocity, and density can be derived conveniently from the present code as constructed. This feature can be expected to minimize the necessary work compared to that of an independent effort.

PRECEDING PAGE NOT FILMED
BLANK

Section 7
REFERENCES

1. K. K. Wang. Aerodynamic Computer Code for Computing Pressure Loading on Complete Missile for Structural Analysis. MDC G7215, MDAC, Huntington Beach, California, January 1978.

2. P. C. Huang and J. P. Matra, Jr. Planform Input Generator (PING), A NASTRAN Preprocessor for Lifting Surfaces-Theoretical Development, User's Manual, and Program Listing. NOLTR-73-199, Naval Ordnance Laboratory, White Oak, Silver Spring, Maryland, December 1973.

3. P. C. Huang and J. P. Matra, Jr. Missile Body Input Generator (BING) A NASTRAN Preprocessor; Theoretical Development, User's Manual, and Program Listing. NSWC-NOL-TR-75-9, Naval Surface Weapons Center, White Oak, Silver Spring, Maryland, March 1975.

4. P. S. Epstein. On the Air Resistance of Projectiles. Proc. Nat. Acad. Sci., U.S.A. 17, 1931; pp. 532-547.

5. C. A. Syvertson and D. H. Dennis. A Second-Order Shock-Expansion Method Applicable to Bodies of Revolution Near Zero Lift. NACA Report 1328, 1957.

6. A. E. Gentry, D. N. Smyth, and W. R. Oliver. The Mark IV Supersonic-Hypersonic Arbitrary Body Program. AFFDL-TR-73-159, 2, AFFDL, Wright-Patterson Air Force Base, Ohio, November 1973.

7. A. G. Hammitt and K. R. Murphy. Approximate Solutions for Supersonic Flow Over Wedges and Cones. JAS, January 1960.

8. D. J. Jones. Tables of Inviscid Supersonic Flow About Circular Cones at Incidence, $\gamma = 1.4$. AGARDograph 137, November 1969.

9. D. J. Jones. A Pressure Formula for an Inclined Circular Cone in Supersonic Flow, $\gamma = 1.4$. AIAA Journal, February 1972; pp. 234-236.

10. E. R. A. Landers. Pressure Distribution on Planar and Cruciform Delta Wings Attached to Cylindrical Bodies in Supersonic Flow, Part II. W. R. E. Report HSA 107, Australian Defense Scientific Service, Weapons Research Establishment, Adelaide, Australia.

11. R. C. Savin. Application of the Generalized Shock-Expansion Method to Include Bodies of Revolution Traveling at High Supersonic Speeds. NACA TN 3345, April 1955.

12. M. H. Bertram and A. Henderson, Jr. Recent Hypersonic Studies of Wings and Bodies. ARS Journal, Vol. 31, No. 8, August 1961.

PRECEDING PAGE NOT FILMED
BLANK

Appendix A

PROGRAM LISTING

PRECEDING PAGE NOT FILMED BLANK

```
      OVERLAY(WOOD,0,0)
      PROGRAM WOODWRD(INPUT=200,OUTPUT=200,TAPE5=INPUT,TAPE6=OUTPUT,
     1           TAPE1=200,TAPE2=100,TAPE3=200,TAPE4=100,TAPE8=100,
     2     TAPE9=100,TAPE11=200,TAPE12,PUNCH=200)
C
C *************************************************************
C *******  SUBSONIC / SUPERSONIC INFLUENCE COEFFICIENT PROGRAM ********
C *************************************************************
C
C              AERODYNAMIC COMPUTER CODE FOR ANLAYSIS OF MISSILE WING,
C              BODY, OR COMBINATIONS.  IT IS BASED ON THE WOODWARD
C              PROGRAM DEVELOPED FOR COMPUTING THE PRESSURE LOADS ON
C              MISSILE. IT READS IN THE GEOMETRY OF MISSILE COMPONENTS
C              AND THE FINITE ELEMENT DESCRIPTION FROM NSWC'S PING OR
C              BING AS INPUT, COMPUTES PRESSURE LOAD AND INTERPOLATES,
C              USING SURFACE FIT,FOR PRESSURE ON THE CENTROID OF EACH
C              ELEMENT.  FOR INFORMATION CONTACT  KENNETH K. WANG
C                                                 MCDONNELL-DOUGLAS
C                                                 HUNTINGTON BEACH, CA.
C
      COMMON/ MAIN / LA, LB, LC, LD, LE, LF, LI,LO, NBODY, NWING, XMACH,
     1           SYM, KACE, NPOLAR, IRW, IPLOT
      COMMON / BODYSP/ DRDX(51), DIDX(51), DRDXC(51), XWROOT, XWTIP
      COMMON /INPUT T/ LIN
      COMMON /DSXB/XBS1,XBS2,DSB1,DSB2
      COMMON /TRNSFR/ IOVR10,IOVR60,IOVR70
      COMMON /ENDJOB/ JOBEND
      COMMON/CORTRN/ NSYM,NTRANS(20),ISYM(20),VA(3,2,20), BTFM(9,2,20)
C
C
C     INITIALIZE
C
      LA=1
      LB=2
      LC=3
      LD=4
      LIN=5
      LO=6
      LF=8
      LE=9
      LI=11
      REWIND1
      REWIND2
      REWIND3
      REWIND4
      REWIND8
      REWIND9
      REWIND11
C
C
C        READ IN CASE SPECIFICATIONS
C
      CALL OVERLAY(4HWANG, 2, 0)
      NPOLAR = 0
C
      IF ( XMACH .LT. 2.5) GO T' 50
C
C        HIGH SUPERSONIC CASE, M.GE. 2
C
      CALL OVERLAY(4HWANG, 7, 0)
      GO TO 1100
C
   50 CONTINUE
C
C        IRW = 0 , DATA ARE NOT SAVED ON TAPE
```

```
C           IRW = 1 , DATA ARE SAVED ON TAPE 12 FOR RESTART
C           IRW = 2 , RESTART RUN, BYPASS ALL AERO CALCULATIONS
C
      IF ( IRW  .LT. 2 ) GO TO 100
      CALL OVERLAY(4HWANG, 7, 0)
      GO TO 1100
  100 CONTINUE
C
      IOVR10 = 0
      CALL OVERLAY(4HWOOD, 1, 0)
C
C
C     CALL AERODYNAMIC LINK
 1050 CALL AERO
C
C
C                DUMMY READ AND WRITE ON TAPE 9 IN AN UNFORMATED FORM TO
C           MOVE SYSTEM ROUTINE  W,SO INTO OVERLAY( 0, 0 )  TO AVOID
C           LOADING ERRORS.
C
C           .......  DO NOT REMOVE  ......
C
C
      IDUM = 1
      IF ( IDUM  .EQ. 1 ) GO TO 1100
      READ  (9)  DUM
      WRITE (9) DUM
C
 1100 CONTINUE
C
      WRITE (6,901)
  901 FORMAT (1H0////*...........END OF COMPUTATIONS .......*)
C
C
C ****************
      STOP
C ****************
      END
C
      SUBROUTINE AERO
C
C .................................................................
C     CONTROL ROUTINE FOR AERODYNAMIC LINKS
C .................................................................
C
      COMMON/ MAIN / NTAPEA,NTAPEB,NTAPEC,NTAPED,NTAPEE,NTAPEF,NTAPEI,
     1          NTAPEO,NBODY,NWING,XMACH,SYM,KACE,NPOLAR,IRW
C
      COMMON /TRNSFR/ IOVR10,IOVR60,IOVR70
      COMMON /ENDJOB/ JOBEND
      COMMON /COM60/ ISAVET
C
C
      REWIND NTAPEA
      REWIND NTAPEB
      REWIND NTAPEC
      REWIND NTAPED
      REWIND NTAPEE
      REWIND NTAPEF
C
C
      CALL OVERLAY(4HWOOD,3,0,6HRECALL)
C
      GO TO ( 500, 1070, 2000), KACE
C
```

```
C          WING ALONE, KACE = 1
C
  500 CONTINUE
C
      IOVR70 = 1
      CALL OVERLAY(4HWANG, 7, 0)
C
      IOVR70 = 4
      CALL OVERLAY(4HWANG, 7, 0)
C
      GO TO 3000
C
C          BODY ALONE, KACE = 2
 1070 CONTINUE
      GO TO 3000
C
C          WING - BODY COMBINATION, KACE = 3
C
 2000 CONTINUE
      IOVR70 = 2
      CALL OVERLAY(4HWANG, 7, 0)
      CALL OVERLAY(4HWOOD, 4, 0,6HRECALL)
C
      IOVR70 = 3
      CALL OVERLAY(4HWANG, 7, 0)
      CALL OVERLAY(4HWOOD, 5, 0,6HRECALL)
C
      IOVR70 = 4
      CALL OVERLAY(4HWANG, 7, 0)
C
 3000 CONTINUE
      IOVR60 = 1
      CALL OVERLAY(4HWOOD, 6, 0,6HRECALL)
C
      IOVR70 = 0
      CALL OVERLAY(4HWANG, 7, 0)
C
C
C ------------------
      RETURN
C ------------------
C
      END
C
      SUBROUTINE FOR EOF (IT)
C
C .....................................................................
C     WRITES E.F. ON A TAPE TO INDICATE THE END OF A DATA SET
C .....................................................................
C
       INTEGER EOF
C
      DATA EOF /4HE.F./
C
      WRITE (IT) EOF
C
C ------------------
      RETURN
C ------------------
      END
C
      SUBROUTINE FSF (NF, NTAPE, IRR)
C
C .....................................................................
C     READS A TAPE UNTIL IT HAS READ THE CHARACTERS E.F.
```

```
C ...................................................................
C
       INTEGER TW
       INTEGER EOF
C
      DATA EOF /4HE.F./
C
      IRR = 0
      K = 0
      DO 1000  I = 1, 10000
      READ(NTAPE) TW
      IF (TW .NE. EOF)  GO TO 1000
      K = K +1
      IF (K .GE. NF)  RETURN
 1000 CONTINUE
      IRR = NF -K
C
C ------------------
      RETURN
C ------------------
      END
C
      FUNCTION INTURP(IDIC,NDIC,LI,LO)
C
C ...................................................................
C     READ AND WRITE A COMMAND CARD,SEE IF COMMAND (IN THE
C     FIRST WORD) IS IN THE DICTIONARY.
C ...................................................................
C
      INTEGER IDIC(1), COM(20), ICOM
C
      EQUIVALENCE (COM(1),ICOM)
C
C
C     IDIC (INSTEAD OF DIC) IS USED AS AN ARGUMENT
C     SO THAT A FIXED-POINT COMPARISON CAN BE MADE
C
      READ (LI,1040)COM
      IF(LO)1010,1010,1000
 1000 WRITE (LO,1050)COM
 1010 DO 1020 INTERP=1,NDIC
      IF (IDIC(INTERP) .EQ. ICOM)  GO TO 1030
 1020 CONTINUE
C
      INTERP=0
 1030 INTURP=INTERP
C
C ------------------
      RETURN
C ------------------
 1040 FORMAT(20A6)
 1050 FORMAT(/1H0,20A6)
      END
C
      OVERLAY(WOOD,1,0)
C/    LIST,NONE
      PROGRAM CGOP
      COMMON/ MAIN / NTAPEA,NTAPEB,NTAPEC,NTAPED,NTAPEE,NTAPEF,NTAPEI,
     1          NTAPEG,NBODY,NWING,IMACH,STM,KACE,NPOLAR,IRW
      COMMON /TRNSFR/ IOVR10,IOVR60,IOVR70
  200 CALL PANEL
C
      END
C
      SUBROUTINE PANEL
```

```
C
C
      COMMON/ MAIN / LA,NTAPEB,NTAPEC,NTAPED,LE,LF,LI,LO,NBODY,NWING,
     1          XMACH, SYM, KACE, NPOLAR, IRW
      COMMON /XYWING/ XW(13,11),YW(11),ZW(11)
      COMMON/DSXB / XBS1, XBS2, DSB1, DSB2
      DIMENSION XB(51),RB(51),ZD(51),THETA(11),X(51),Y(11),Z(11),BC(10),
     1XBAR(50),YBAR(10),ZBAR(10),XC(50),THET(11),CHORD(50),AREA(50,10),
     2KW(10),ZBR(110,10),XBR(110,10),YBR(110,10),CW(110,10),THETW(10)
     3,ARW(110,10),XCW(110,10)
C
      REWIND NTAPEC
      XCPT = 0.95
      I=1
      ZO=0.
      ON=1.
      IF ( NBODY ) 100, 100, 10
   10 READ (LI,1010) BODYS, BTHET, PLANE, RP, YAW
      NBODYS=BODYS
      NTHET=BTHET
      NPLANE=PLANE
      NXLE=0
      READ  (LI,1010) ( XB(N), N=1,NBODYS)
      READ  (LI,1010) ( RB(N), N=1,NBODYS)
      READ  (LI,1010) ( ZD(N), N=1,NBODYS)
      READ  (LI,1010) ( THETA(N), N=1,NTHET )
      READ  :LI,1010) XBS1, XBS2, DSB1, DSB2
      DO 15 N=1,NTHET
   15 THETA(N)=THETA(N)/57.2957795
      !F(PLANE) 300,300,20
C
C         BODY-WING CASE, REQUIRES PANELED BODY
C
   20 CONTINUE
      PEAD  (LI,1010) ( X(N), N=1,NPLANE)
      ALPHAA=0
      NPLN1=NPLANE-1
      NROW=NTHET-1
C
C
      IF(RP) 22,22,28
C
C         NON-CIRCULAR BODY
C
   22 CONTINUE
      READ  (LI,1010) ( Y(N), N=1,NTHET)
      READ  (LI,1010) ( Z(N), N=1,NTHET)
      GO TO 35
C
C         CURCULAR BODY
C
   28 DO 30 N=1,NTHET
      Y(N)=RP*SIN(THETA(N))
   30 Z(N)=RP*COS(THETA(N))
 35   DO 40 N=1,NROW
      YBAR(N)=(Y(N)+Y(N+1))/2.
      ZBAR(N)=(Z(N)+Z(N+1))/2.
   40 BC(N)=SQRT((Y(N+1)-Y(N))**2+(Z(N+1)-Z(N))**2)
      DO 50 M=1,NPLN1
      XBAR(M)=(X(M+1)+X(M))/2.
      CHORD(M)=X(M+1)-X(M)
   50 XC(M)=X(M)+XCPT*CHORD(M)
      DO 60 N=1,NROW
   60 THET(N)=ATAN2((Z(N+1)-Z(N)),(Y(N+1)-Y(N)))
      WRITE(LO,900)
      WRITE(LO,910)
      NP=0
      DO 70 N=1,NROW
      DO 70 M=1,NPLN1
      NP=NP+1
      WRITE(LO,920)  NP,I,X(M),Y(N),Z(N),X(M),Y(N+1),Z(N+1),
     1X(M+1),Y(N),Z(N),X(M+1),Y(N+1),Z(N+1)
   70 WRITE (NTAPEC) NP,I,X(M),Y(N),Z(N),X(M),Y(N+1),Z(N+1),
     1X(M+1),Y(N),Z(N),X(M+1),Y(N+1),Z(N+1)
      WRITE(LO,960)
      WRITE(LO,950)
      NP=0
      DO 80 N=1,NROW
      DO 80 M=1,NPLN1
      NP=NP+1
      AREA(M,N)=CHORD(M)*BC(N)
      WRITE(LO,960)  NP,XBAR(M),YBAR(N),ZBAR(N),XC(M),YBAR(N),ZBAR(N),
     1AREA(M,N),THET(N  ),Z0,CHORD(M)
   80 WRITE (NTAPEC) NP,XBAR(M),YBAR(N),ZBAR(N),XC(M),YBAR(N),ZBAR(N),
     1AREA(M,N),THET(N  ),Z0,CHORD(M)
      WRITE (NTAPEC) NPLN1,XCPT
      NBODY=NP
      IF(NWING)500,500,120
  100 IF(NWING)500,500,110
C
C          BODY ONLY OR BODY-WING CASE
C
  110 CONTINUE
      NBODY=0
      PLANE=0.
  120 NP=0
      WRITE(LO,970)
      WRITE(LO,980)
      DO 200 L=1,NWING
      CALL WINGP(NB,NC,Y0,ZA,DIN)
      THETW(L)=DIN
      WRITE(LO,1020) L,Y0,ZA,DIN
      NB1=NB-1
      NC1=NC-1
      KW(L)=NB1*NC1
      DO 150 N=1,NB1
      DO 150 M=1,NC1
      NP=NP+1
      WRITE (NTAPEC) NP,I,XW(M,N),YW(N),ZW(N),XW(M,N+1),YW(N+1),ZW(N+1),
     1XW(M+1,N),YW(N),ZW(N),XW(M+1,N+1),YW(N+1),ZW(N+1)
      WRITE(LO,920)  NP,I,XW(M,N),YW(N),ZW(N),XW(M,N+1),YW(N+1),ZW(N+1),
     1XW(M+1,N),YW(N),ZW(N),XW(M+1,N+1),YW(N+1),ZW(N+1)
      K=M+(N-1)*NC1
      CR=XW(M+1,N)-XW(M,N)
      CT=XW(M+1,N+1)-XW(M,N+1)
      RY=(1.+CT/(CR+CT))/3.0
      RY1=1.0-RY
      XWL=RY*XW(M,N+1)+RY1*XW(M,N)
      XWT=RY*XW(M+1,N+1)+RY1*XW(M+1,N)
      CW(K,L)=XWT-XWL
      YDUM=SQRT((YW(N+1)-YW(N))**2+(ZW(N+1)-ZW(N))**2)
      ARW(K,L)=YDUM*.5*(CR+CT)
      XBR(K,L)=(XWL+XWT)/2.
      YBR(K,L)=YW(N+1)*RY+YW(N)*RY1
      ZBR(K,L)=ZW(N+1)*RY+ZW(N)*RY1
  150 XCW(K,L)=XCPT*XWT+(1.0-XCPT)*XWL
  200 CONTINUE
      WRITE(LO,990)
      WRITE(LO,1000)
      NP=0
      DO 250 L=1,NWING
      KL=KW(L)
      DO 250 K=1,KL
      NP=NP+1
      WRITE(LO,960) NP,XBR(K,L),YBR(K,L),ZBR(K,L),XCW(K,L),YBR(K,L),
     1ZBR(K,L),ARW(K,L),THETW(L),ZO,CW(K,L)
  250 WRITE(NTAPEC) NP,XBR(K,L),YBR(K,L),ZBR(K,L),XCW(K,L),YBR(K,L),
     1ZBR(K,L),ARW(K,L),THETW(L),ZO,CW(K,L)
      NWING=NP
      WRITE (NTAPEC) NC1,XCPT
      IF  (PLANE.GT.O.) WRITE(NTAPEC) NTHET
      IF  (PLANE.GT.O.) WRITE(NTAPEC) (THETA(N),N=1,NTHET)
      IF(PLANE) 500,5C0,400
  300 NBODY=-1
      REWIND NTAPED
      WRITE(NTAPED) NDUM,NDUM,NDUM,NTHET
      WRITE(NTAPED) (THETA(N),N=1,NTHET)
      REWIND NTAPED
  400 REWIND NTAPEB
      WRITE (NTAPEB) ON,ON,ZO,ZA,DSB1,DSB2
      WRITE (NTAPEB) NBODYS,NXLE
      WRITE (NTAPEB) (XB(N),RB(N),ZD(N),N=1,NBODYS)
      REWIND NTAPEB
  500 REWIND NTAPEC
      NPANEL=NWING+NBODY
      RETURN
  900 FORMAT (1H1,9X,*BODY PANEL CORNER POINT LOCATIONS EXPRESSED IN *,
     A          *THE AERO COORDINATE SYSTEM*/10X,
     186H1 AND 2 INDICATE BODY PANEL LEADING-EDGE POINTS, 3 AND 4 INDICA
     2TE TRAILING-EDGE POINTS)
 910  FORMAT (1H0,5X,5HPANEL,   4(8X,1HX,8X,1HY,8X,1HZ)/
     15X,2HNO,2X,5HPARTS,6X,
     2    3(1H1,8X),3(1H2,8X),3(1H3,8X),3(1H4,8X)//)
 920  FORMAT (1H ,3X,I3,3X,I1,3X,12F9.3)
 930  FORMAT (1H ,13X,12F9.5)
  940 FORMAT (1H1,2X,*BODY PANEL CENTROID AND CONTROL POINT LOCATIONS*,
     1          * EXPRESSED IN THE AERO COORDINATE SYSTEM*)
 950  FORMAT (1H0,5HPANEL,6X,2(1HX,8X,1HY,8X,1HZ,8X),5X,4HAREA,9X,
     16HTHETA-,6X,6HALPHA-, 5X,5HCHORD/13X,3(1HC,8X),3(2HCP,7X),
     217X,6HINCLIN,6X,5HINCID//)
 960  FORMAT (1H ,1X,I3,4X,6F9.3,5X,F9.3,5X,F12.3,2F9.3)
  970 FORMAT (1H1,9X,*WING PANEL CORNER POINT LOCATIONS EXPRESSED IN *,
     A          *THE AERO COORDINATE SYSTEM*/10X,
     186H1 AND 2 INDICATE WING PANEL LEADING-EDGE POINTS, 3 AND 4 INDICA
     2TE TRAILING-EDGE POINTS)
  980 FORMAT (1H0,5X,5HPANEL,   4(8X,1HX,8X,1HY,8X,1HZ)/
     15X,2HNO,2X,5HPARTS,6X,
     2    3(1H1,8X),3(1H2,8X),3(1H3,8X),3(1H4,8X)//)
  990 FORMAT (1H1,9X,*WING PANEL CENTROID AND CONTROL POINT LOCATIONS *,
     A          *EXPRESSED IN THE AERO COORDINATE SYSTEM*)
 1000 FORMAT (1H0,1X,5HPANEL,6X,2(1HX,8X,1HY,8X,1HZ,8X),5X,4HAREA,12X,
     16HTHETA-,3X,6HALPHA-, 5X,5HCHORD/14X,3(1HC,8X),3(2HCP,7X),
     220X,5HDIHED,4X,6HCAMBER,4X//)
 1010 FORMAT(7F10.0)
 1C20 FORMAT(1H0,4X,9HWING NO. ,I2,5X,9H YPIVOT =,F10.3,5X,9H ZPIVOT =,
     1F10.3,5X,16H WING DIHEDRAL =,F10.3,1X,7HRADIANS,1X/)
      END
C
      SUBROUTINE WINGP(NB,NC,YO,ZA,DIN)
      COMMON/ MAIN / LA,NTAPEB,NTAPEC,NTAPED,LE,LF,LI,LO,NBODY,NWING,
     1          XMACH, SYM, KACE, NPOLAR, IRW
      COMMON /XYWING/ XY(13,11),YW(11),ZW(11)
      DIMENSION XW(12),DY(11)
      READ (LI,1010) XO,YO,CR,A,B,C
      READ (LI,1010) XNB,XNC,CASE,ZA
      NB=XNB
      NC=XNC
      NCA=CASE
      PERCT = 0.1
C
      CRR = CR - XO
      DCR = ( 1.0 - PERCT ) * CRR / ( XNC - 2.0 )
      XW(1)=XO
      XW(2) = XW(1) + PERCT* CRR
      DO 145 M=3,NC
      XW(M) = XW(M-1) + DCR
  145 CONTINUE
      DY = B - YO
      DBY=DY/(XNB-1.)
      YW(1)=YO
      DO 165 N=2,NB
      YW(N) = YW(N-1) + DBY
  165 CONTINUE
      SLE=(A-XO)/(B-YO)
      STE=(C-CR)/(B-YO)
      IF(SLE-STE)210,230,210
  210 CONTINUE
      DX = ( C-A ) / ( CR - XO )
      DO 215 M=1,NC
      XT(M,NB) = A + DX*( XW(M) - XO )
  215 CONTINUE
      DO 220 N=1,NB
      DY(N)=(YW(N)-YO)/(B-YO)
      DO 220 M=1,NC
      XT(M,N) = XW(M) + ( XT(M,NB) - XW(M) ) * DY(N)
  220 CONTINUE
      GO TO 250
  230 DO 240 N=1,NB
      DY(N)=YW(N)-YO
      DO 240 M=1,NC
      XT(M,N) = XW(M) + SLE*DY(N)
  240 CONTINUE
  250 CONTINUE
      CALL WROTAT(NB,YO,ZA,DIN)
      RETURN
 1010 FORMAT(7F10.0)
      END
C
      SUBROUTINE WROTAT(NB,YO,ZA,WNGDIN)
      COMMON/ MAIN / LA, LB, LC, LD, LE, LF, LI,LO, NBODY, NWING, XMACH,
     1         SYM, KACE, NPOLAR, IRN
      COMMON /XYWING/ XT(13,11),YW(11),ZW(11)
      DIMENSION DYPIVT(11)
      WNGDIN=0.
      READ (11,1020)  PIVOT, WNGDIN, WANG
      YPIVOT=YO
      ZPIVOT=ZA
C  PIVOT, DIHEDRAL INDICATOR
C  YPIVOT, ZPIVOT, PIVOT POINT
C  WNGDIN, DIHEDRAL ANGLE
      DO 50 I=1,NB
   50 ZW(I)=ZA
C
      IF (PIVOT .LT. 0.01 ) RETURN
      DO 100 I=1,NB
      DYPIVT(I)=YW(I)-YPIVOT
 100  CONTINUE
      WNGDIN=WNGDIN/57.2957795
      COSDIN=COS(WNGDIN)
      SINDIN=SIN(WNGDIN)
      DO 200 I=1,NB
      YW(I)=YPIVOT + DYPIVT(I)*COSDIN
      ZW(I)=ZPIVOT + DYPIVT(I)*SINDIN
  200 CONTINUE
 1020 FORMAT (7F10.4)
      RETURN
      END
C
      OVERLAY(WANG, 2, 0)
      PROGRAM PRELIM
C
C
C              READ INPUT,  WRITE  ON TAPE 11 FOR SUBSEQUENT COMPUTATION
C
      COMMON/ MAIN / LA, LB, LC, LD, LE, LF, LI,LO, NBODY, NWING,XMACHR,
     1         SYMM, KACE, NPOLAR, IRW, IPLOT
      COMMON / BODYSP/ DRDX(51), DZDX(51), DRDXC(51)
      COMMON/DEFAU / BODY, BODYS, BTHET, CPCALC,WCASE, NRUN, DADEG, ARB,
     1          PLANE, POLAR, RP, XNACEL, SYM, THICK,  WING, XMACH,
     2          XNBB,XNCC,PINF,IPUNCH,SID(20), CKASE, ARW(10), XMOM,
     3          YAW
      COMMON/ BODY2 / NBPFL, NBODYS, NTHET, XB(51), RB(51), ZB(51),
     1          THETA(11), RPP, XBPANL(20)
      COMMON/ NACEL /NACELP, XN(51,2), RN(51,2), ZN(51,2), ARN(2)
      COMMON/CORTRN/ NSYM,NTRANS(20),ISYM(20),VA(3,2,20), BTFM(9,2,20)
      COMMON/INPUT / TX(2, 50)
      COMMON/ POINTP/ TY(4), TYP(4), TYPP(4)
      COMMON/ POINTS / NPTS, NVAR
      COMMON/WINGFM/ XWI(30,20), YWI(30,20), ZWI(30,20),NWPI(20),NCHORD,
     1          NSPAN, ISOLID, NPLANE, ICAMBR, ITWIST,ITHICK,IFORM(10)
     2          , PIVOT(10), DIHED(10), NWPANL, XLG(10), XTG(10),
     3          BROLL
      INTEGER TITLE(20)
      DIMENSION NBC(2, 6)
      DATA NBC / 4, 9, 4, 6, 3, 6, 3, 6, 3, 4, 2, 4 /
C
      NAMELIST/ UNIFID /ICASE,NTRANS,IPUNCH,   IRW, POLAR,ISOLID,   SID,
     1           NBPFL,    XB,     RB,     ZB,     RP,  NWPI,    YAW, BROLL,
     2           NWING, IFORM, PIVOT, THICK, DIHED,    XWI,    YWI,    ZWI,
     3          XNACEL,NACELP,     XN,     RN,    ZN,NWPANL,    XLG,    XTG,
     4           XMACH,  PINF, DADEG,    ARB,    ARW,    ARN
C
C              READ IN DESCRIPTIVE TITLE
C
      READ (5,1000)  TITLE
      WRITE (6,1010) TITLE
C
C              SET DEFAULT VALUES
C
      CALL DEFAUT
C
C              READ NAMELIST INPUT FOR UNIFIED COMPUTER CODE
C
      READ ( 5, UNIFID )
      WRITE ( 6, UNIFID )
C
      KACE = ICASE
C
      XMACHR = XMACH
      YAW = YAW / 57.2957795
      BROLL = BROLL / 57.2957795
      NSTR = NWING - 2
      IF ( NWING  .LE. 2 ) GO TO 30
      XNBB = FLOAT( NBC( 1, NSTR ) )
      XNCC = FLOAT( NBC(2,NSTR) )
```

```
   30 CONTINUE
      XNB = XNBB
      XNC = XNCC
      NB = XNB
      NC = XNC
      NBODY = 0
      NCHORD = NC
      NSPAN = NB
C
C        FIRST RUN IRW .LT. 2
C        RESTART RUN IRW = 2
C
      IF ( IRW  .EQ. 2 ) GO TO 500
C
C        READ IN COORDINATE SYSTEM SPECIFICATION FOR BODY AND WING AND
C        COMPUTE TRANSFORMATION MATRICES.
C
      IF ( ICASE  .NE. 1 ) NBODY = 1
      ANACEL = ABS( XNACEL )
      NACELS= IFIX(ANACEL)
      NW2 = 0
      IF ( ICASE  .EQ. 2 ) GO TO 45
      DO 40 I=1,NWING
      NCORD = 1
      IF ( IFORM(I) .EQ. 1 ) NCORD = 2
      NW2 = NW2 + NCORD
   40 CONTINUE
   45 CONTINUE
      NS = NBODY + NW2 + NACELS
      DO 50 I=1,NS
      NSYM = I
      CALL CORDTR
   50 CONTINUE
C
      IF ( ICASE  .EQ. 2 ) GO TO 70
C
C        TRANSFORM WING COORDINATES TO THAT OF AERO
C
      NSYM = NBODY
      DO 60 I = 1,NWING
      NSYM = NSYM + 1
      NW = 2 * I - 1
      CALL WTRANS( NW, I )
      CALL CORTRN( NW, I )
      IF ( IFORM(I)  .NE. 1 ) GO TO 60
      NSYM = NSYM + 1
      NW = NW + 1
      CALL WTRANS( NW, I )
      CALL CORTRN( NW, I )
C
   60 CONTINUE
      IF ( NACELS  .EQ. 0 ) GO TO 70
C
C        TRANSFORM THE NACEL AXIS XN TO THAT OF THE BODY AXIS
C
      DO 51 K = 1, NACELS
      NSYM = 1 + NW2 + K
      NT = NTRANS( NSYM )
      DO 51 ITRAN = 1, NT
      DO 51 J = 1, NACELP
      XN(I) = XN(J) + VA( 1, ITRAN, NSYM )
   51 CONTINUE
C
   70 CONTINUE
C
```

```
C          FOR HIGH SUPERSONIC CASE, BYPASS THE GEOMETRY AND AERO
C         ANALYSIS SECTION
C
      IF ( XMACH .LT. 2.5 ) GO TO 80
C
      CALL OVERLAY(4HWANG, 2, 1)
C
      GO TO 500
   80 CONTINUE
C
C        PREPARE INPUTS AND WRITE ON TAPE 11 FOR AERO CODE
C
C        ICASE = 1 ,  WING  ONLY
C        ICASE = 2 ,  BODY  ONLY
C        ICASE = 3 ,  COMPLETE MISSILE
C
      GO TO ( 100, 200, 300 ) , ICASE
C
C        WING ONLY CASE
C
  100 CONTINUE
      KACE = 1
      CALL WINGIN
      GO TO 500
C
  200 CONTINUE
C
C        BODY ONLY CASE
C
      KACE = 2
      NBODY = 1
      XNACEL = 0.0
      NWING = 0.0
      PLANE = 0.0
      CALL BODYIN
      GO TO 500
C
  300 CONTINUE
C        COMPLETE MISSILE CASE
C
      KACE = 3
      CALL BANWIN
C
  500 CONTINUE
C
C
 1000 FORMAT(20A4)
 1010 FORMAT(1H1,20A4)
 1020 FORMAT (7F10.0)
C
C
      END
C
      SUBROUTINE BODYPN(XLAST)
C
C
C           PANEL BODY SECTION TO CORRESPONDING WITH THE WING PANEL FOR
C           AERODYNAMIC COMPUTATIONS
C
      COMMON/ MAIN / LA, LB, LC, LD, LE, LF, LI,LO, NBODY, NWING, XMACH,
     1         SYM, KACE, NPOLAR, IRW
      COMMON/ BODY2 / NSPFL, NBODYS, NTHET, XB(51), RB(51), ZB(51),
     1         THETA(11), RP, XN(20)
      COMMON / BODYSP/ DRDX(51), DZDX(51), DRDXC(51)
      COMMON/CORTRN/ NSYM,NTRANS(20),ISYM(20),VA(3,2,20), BTFM(9,2,20)
      COMMON/DEFAU / BODY, BODYS, BTHET, CPCALC, NCASE, NRUN, DADEG, ARB
     1         ,PLANE, POLAR, RPP,XNACEL, SYMM, THICK, WING, XMACHN, XNB
     2         , XNC, PINF, IPUNCH, CKASE, ARW(10), XMOM,
     3         YAW
      COMMON/WINGFM/ XWI(30,20), YWI(30,20), ZWI(30,20),NWPI(20),NCHORD,
     1         NSPAN, ISOLID, NPLANE, ICAMBR, ITWIST,ITHICK,IFORM(10)
     2         , PIVOT(10), DIH(10), NWPANL, XLG(10), XTG(10)
      COMMON/POINTP/ TY(4), TYP(4), TYPP(4)
C
      XCPT = 0.95
      PERCT = 0.1
      NC = XNC
      NTHET = BTHET
      NCK = NWPANL + NTHET * ( NC - 1 )
      IF ( NCK .GT. 100) NC = NC / 2
      XNC = FLOAT( NC )
      NC2 = NC - 2
      IF ( NC2 .LE. 0 ) NC2 = 1
      NPC = 1
C
C
C          DETERMINE BODY PANEL RADIUS  RP
C
      YMIN = YWI( 1,1)
      NW2 = 2 * NWING
      XMIN = XWI( 1, 1)
      XMAX = XWI( 2,1)
      IF ( NWING  .EQ. 1 ) GO TO 205
      DO 200 I=3,NW2,2
      J = I
      NW1 = I/2 + 1
      IF ( IFORM(NW1) .EQ. 2 ) J = NW1
      YMIN = AMIN1( YWI(1,J), YMIN )
  200 CONTINUE
C
  205 CONTINUE
      RP = YMIN
      RPP = RP
C
C        PANEL BODY USING INPUT OF NWPANL, XLG, XTG
C
      WSPAN = XTG(1) - XLG(1)
      DX1 = 0.2 * WSPAN
      DX2 = ( 1.0 - PERCT ) * WSPAN / NC2
      XN(1) = XLG(1) - DX1
      XN(2) = XLG(1)
      XN(3) = XLG(1) + PERCT * WSPAN
      IC = 3
      DO 300 I=3,NC
      IC = IC + 1
      XN(IC) = XN(IC-1) + DX2
  300 CONTINUE
C
  305 CONTINUE
      NPC = NPC + 1
      IF ( NPC .GT. NWPANL ) GO TO 700
      IF ( ABS( XLG(NPC) -XTG(NPC-1) ) .LE. 1.0E-3 ) GO TO 310
      IC = IC + 1
      XN(IC) = XN(IC-1) + 0.5*( XLG(NPC) - XTG(NPC-1) )
      IC = IC + 1
      XN(IC) = XLG(NPC)
  310 CONTINUE
      WSPAN = XTG(NPC) - XLG(NPC)
      DX2 = ( 1.0-PERCT ) * WSPAN / NC2
      IC = IC + 1
      XN(IC) = XLG(NPC) + PERCT * WSPAN
      DO 350 I =3, NC
      IC = IC + 1
      XN(IC) = XN(IC-1) + DX2
  350 CONTINUE
C
      IF ( NPC .LT. NWPANL ) GO TO 305
  700 CONTINUE
C
C           ADD ONE RING OF PANEL ON BODY BEYONDTHE WING TRAILING EDGE
C
      DXLAST = 0.5*( XLAST - XN(IC) )
      DXLAST = AMIN1( DXLAST, DX2)
      IC = IC + 1
      XN(IC) = XN(IC-1) + DXLAST
      NPLANE = IC
C
C           CALCULATE DRDX AT THE CONTROL POINT
C
      NP1 = NPLANE - 1
      DO 400 I = 1, NP1
      TY(1) = XN(I) + XCPT*( XN(I+1) - XN(I) )
      CALL POINT(1)
      DRDXC(I) = TYP(2)/TYP(1)
  400 CONTINUE
C
      RETURN
      END
C
      SUBROUTINE CORTRN( NW, IW )
C
C                 TRANSFORM COORDINATES FROM THE LOCAL SYSTEM TO THE AERO
C            SYSTEM  USING THE TRANSFORM MATRIX BTFM
C
C
      COMMON/CORTRN/ NSYM,NTRANS(20),ISYM(20),VA(3,2,20), BTFM(9,2,20)
      COMMON/WINGFM/ XWI(30,20), YWI(30,20), ZWI(30,20),NWPI(20),NCHORD,
     1          NSPAN, ISOLID, NPLANE, ICAMBR, ITWIST,ITHICK,IFORM(10)
     2          , PIVOT(10), DIM(10), NWPANL, XLG(10), XTG(10)
      DIMENSION TX(3), X(3)
C
      NWP = NWPI( NW )
      IF ( IFORM( IW )  .EQ. 2 ) NWP = 4
      NT = NTRANS( NSYM )
      DO 250 ITRAN = 1, NT
C
      DO 220 L = 1, NWP
      X(1) = XWI( L, NW )
      X(2) = YWI( L, NW )
      X(3) = ZWI( L, NW )
C
      DO 100 J=1,3
      SUM = 0.0
      DO 120 K=1,3
      JK = K + ( J-1) * 3
      SUM = SUM + X(K) * BTFM( JK, ITRAN, NSYM )
  120 CONTINUE
      TX(J) = SUM
  100 CONTINUE
C
      XWI(L,NW) = TX(1) + VA(1, ITRAN, NSYM )
      YWI(L,NW) = TX(2) + VA(2, ITRAN, NSYM )
      ZWI(L,NW) = TX(3) + VA(3, ITRAN, NSYM )
  220 CONTINUE
C
```

```
  250 CONTINUE
C
      RETURN
      END
C
      SUBROUTINE BODYPF( XB, RB, ZD, THETA, NP, NTHET, NBODYS, XBS1,
     1         XBS2, DSB1, DSB2, XLAST)
C
C        GENERATES BODY PROFILE R = R(X), ZD = ZD(X) AND MERIDIAN ANGLE
C              THETA.
C        COMPUTES SLOPES DSB1, DSB2
C
C
C
      COMMON/ MAIN / LA, LB, LC, LD, LE, LF, LI,LO, NBODY, NWING,XMACHN,
     1         SYMM, KACE, NPOLAR, IRW
      COMMON / BODYSP/ DRDX(51), DZDX(51), DRDXC(51)
      COMMON/INPUT / TX(2, 50)
      COMMON/ POINTP/ TY(4), TYP(4), TYPP(4)
      COMMON/ POINTS/ NPTS, NVAR
      DIMENSION XB(51), RB( 51), ZD( 51), THETA( 11)
C
      NPTS = NP
      XLAST = XB(NP)
C
C        SPLINE FIT INPUT PROFILE AFTER COORDINATE TRANSFORM.
C
      ICAMBR = 0
      DO 100 I=1,NPTS
      TX(1,I) = XB(I)
      TX(2,I) = RB(I)
      IF ( ZD(I)  .LT. 1.E-10 ) GO TO 100
      ICAMBR = 1
      TX(3,I) = ZD(I)
  100 CONTINUE
C
      NVAR = 3
      IF ( ICAMBR  .EQ. 0 ) NVAR = 2
      CALL SPLFIT
      IF ( XMACHN .GE. 2.50 ) GO TO 260
C
C        FOR IMPROVED ACCURACY, FIRST 20 PERCENT OF BODY IS DIVIDED INTO
C        10 EQUAL SECTIONS, THE REMAINING PART IS DIVIDED INTO 15 EQUAL
C        SECTIONS.
      N1 = 11
      XMID  = XB(NP)
      NBD1 = NBODYS - 1
      DX = XMID * 0.2 / 10.0
      DO 200 I=2,N1
      AI = I - 1
      TY(1) = DX * AI
      XB(I) = TY(1)
      CALL POINT(1)
      RB(I) = TY(2)
      IF ( ICAMBR  .EQ. 0 ) GO TO 200
      ZD(I) = TY(3)
  200 CONTINUE
C
      NX1 = N1 - 1
      N2 = N1 + 1
      DX = ( XLAST - XB(N1) ) / ( NBD1 - NX1 )
      DO 250 I=N2, NBODYS
      XB(I) = XB(I-1) + DX
      TY(1) = XB(I)
      CALL POINT(1)
      RB(I) = TY(2)
      IF( ICAMER  .EQ. 0 ) GO TO 250
      ZD(1) = TY(3)
  250 CONTINUE
      GO TO 276
C
C             FOR HIGH SUPERSONIC CASE, XB, RB EQAULLY DIVIDED
C
  260 CONTINUE
      DX = XLAST/(NBODYS-1)
      DO 265 I =2, NBODYS
      XB(I) = (I-1)*DX
      TY(1) = XB(I)
      CALL POINT(1)
      RB(I) = TY(2)
      ZD(I) = TY(3)
  265 CONTINUE
  276 CONTINUE
C
C
C          CALCULATE THE SLOPE AND CAMBER OF BODY AT POINT MIDWAY BETWEEN
C                 STATION.
C
      DO 275 I=2,NBODYS
      TY(1) = ( XB(I-1) + XB(I) ) * 0.5
      CALL POINT(1)
      DRDX(I-1) = TYP(2) / TYP(1)
      IF ( ICAMBR  .EQ. 0 ) GO TO 275
      DZDX(I-1) = TYP(3) / TYP(1)
  275 CONTINUE
      NBS1 = NBODYS - 1
      DRDX(NBODYS) = (RB(NBODYS) - RB(NBS1) )/ ( XB(NBODYS) - XB(NBS1) )
C
      TY(1) = XB(1)
      CALL POINT(1)
      DSB1 = TYP(2) / TYP(1)
      TY(1) = XB(51)
      CALL POINT(1)
      DSB2 = TYP(2) / TYP(1)
C
C           GENERATES THETA FOR BODY PANEL
C
      DTH = 180.0 / FLOAT(NTHET - 1)
C
      DO 300 I=1,NTHET
      AI = I - 1
      THETA(I) = DTH * AI
  300 CONTINUE
C
      XBS1 = 1.0
      XBS2 = NBODYS
C
      RETURN
      END
C
C/    LIST,NONE
      SUBROUTINE DEFAUT
C
      COMMON/DEFAU / BODY, BODYS, BTHET, CPCALC,NCASE, NRUN, DADEG, ARB,
     1         PLANE, POLAR, RP, XNACEL, SYM, THICK,  WING, XMACH,
     2         XNB, XNC, PINF, IPUNCH, SID(20), CKASE, ARW(10), XNON,
     3         YAW
C
      BANK = 0.0
      BODY = 0.0
      BODYS = 25.0
      BTHET = 7.0
      CPCALC = 1.0
      DADEG = 0.0
      DIM = 0.0
      NRUN = 1.
      IPUNCH = 0
      CKASE = 2
      PINF = 14.7
      PIVOT = 0.0
      POLAR = 0.0
      RFAREA = 0.0
C
      RP = 0.0
      DO 100 I=1,20
      SID(I) = 1.0
  100 CONTINUE
      SYM = 1.0
      THICK = 1.0
      THKOMP = 1.0
      TWIST = 0.0
      WING = 0.0
      WCASE = 3
      XMACH = 2.0
      XMACEL = 0.0
      XNB = 6.0
      XNC = 9.0
C     XNB = 3.0
C     XBC = 4.0
      XP = 0.0
      ZA = 0.0
      ZP = 0.0
      YAW = 0.0
      XMOM = 0.0
C
      RETURN
      END
C
      SUBROUTINE CORDTR
C
C
C              READ IN COORDINATE SPECIFICATIONS VECTORS VA, VB, AND VC
C         AND COMPUTES THE TRANSFORMATION MATRICES  BTFM  FOR EACH
C         COORDINATE SYSTEM
C
      COMMON/CORTRN/ NSYM,NTRANS(20),ISYM(20),VA(3,2,20), BTFM(9,2,20)
C
      DIMENSION ICOORD(3), VB( 3, 5), VC( 3, 5)
C
      DATA ICOORD/1HR, 1HC, 1HS /
C
C
C              COMPUTE COORDINATE SYSTEM TRANSFORM MATRICES
C
      NTM = NTRANS(NSYM)
      DO 100 ITRAN = 1, NTM
C
      READ (5,901) ID, ICHECK,NN, (VA(I,ITRAN,NSYM), I=1,3),(VB(I,ITRAN)
     1       , I=1,3)
      READ (5,902) (VC( I, ITRAN), I=1,3)
      IF ( ITRAN  .GT. 1 ) GO TO 50
      IF ( ICOORD(1) .EQ. ICHECK ) ISYM( NSYM ) = 0
      IF ( ICOORD(2) .EQ. ICHECK ) ISYM( NSYM ) = 1
      IF ( ICOORD(3) .EQ. ICHECK ) ISYM( NSYM ) = 2
C
   50 CONTINUE
```

```
      CALL TRANSM( VA(1,ITRAN,NSYM),VB(1,ITRAN),VC(1,ITRAN),BTFM(1,
     1           ITRAN, NSYM)   )
  100 CONTINUE
C
C
  901 FORMAT ( A5, A1, 2X, I8, 8X, 6F8.2 )
  902 FORMAT ( 8X, 3F8.2)
C
      RETURN
      END
C
      SUBROUTINE BODYIN
C
C
      COMMON/ MAIN / LA, LB, LC, LD, LE, LF, LI,LO, NBODY, NWING,XMACHN,
     1          SYMM, KACE, NPOLAR, IAW
      COMMON/ BODY2 / NBPFL, NBODYS, NTHET, XB(51), RB(51), ZB(51),
     1           THETA(11), RPP, XBPANL(20)
      COMMON / BODYSP/ DRDX(51), DZDX(51), DRDXC(51)
      COMMON/CORTRN/ NSYP,NTRANS(20),ISYM(20),VA(3,2,20), BTFM(9,2,20)
      COMMON/DEFAU / BODY, BODYS, BTHET, CPCALC,NCASE, NRUN, DADEG, ARB,
     1           PLANE, POLAR, RP, XMACEL, SYM, THICK,  WING, XMACH,
     2           XMB, XMC, PINF, IPUNCH, SID(20), CKASE, ARM(10), XMOM,
     3           YAW
      COMMON/WINGFM/XWI(30,20),YWI(30,20),ZWI(30,20),NWPI(20),NCHORD,
     1           NSPAN,ISOLID,NPLANE,ICAMBR,ITWIST,ITHICK,IFORM(10),
     2             PIVOT(10), DIN(10), NWPANL, XLG(10), XTG(10)
C
      NBODY = 1
      NBODYS = IFIX( BODYS )
      NSYM = 1
      NTHET = IFIX( BTHET )
      NWING = 0
      PLANE = 0.0
      THKOMP = 1.0
      WING = 0.0
      SOLID = 0.0
      THICK = 0.0
      PUNCH = FLOAT( IPUNCH )
C
C
C          READ IN BODY PROFILE AND COMPUTE BODY RADIUS DISTRIBUTION AT
C          OPTIMUM LOCATIONS.
C
      CALL BODYPF( XB, RB, ZB, THETA, NBPFL, NTHET, NBODYS, XBS1, XBS2,
     1           DSB1, DSB2, XLAST)
C
      BODYS = FLOAT( NBODYS )
C
C          WRITE ON TAPE 11 FOR AERODYNAMIC ANALYSIS.
C
      REWIND  11
      WRITE (11,1020) BODYS, BTHET, PLANE, RP, YAW
      WRITE ( 11,1020) ( XB(N), N=1,NBODYS)
      WRITE ( 11,1020) ( RB(N), N=1,NBODYS)
      WRITE ( 11,1020) ( ZB(N), N=1,NBODYS)
      WRITE ( 11,1020) ( THETA(N), N=1,NTHET)
      WRITE (11,1020) XBS1,XBS2,DSB1,DSB2
      WRITE (11,1020) XMACH, SYM, THKOMP,YAW
      WRITE (11,1020) CKASE,CPCALC,POLAR,THICK,CAMB,XMACEL,PINF,PUNCH,
     1          SOLID, WING, ( SID(K), K=1,20), (IFORM(K), K=1,10), XMOM
      WRITE ( 11,1020) ARB, DADEG
      REWIND 11
C
 1020 FORMAT (7F10.4)
```

```
C
      RETURN
      END
C
      SUBROUTINE BANWIN
C
C
      COMMON/ MAIN / LA, LB, LC, LD, LE, LF, LI,LO, NBODY, NWING,XMACHN,
     1          SYMM, KACE, NPOLAR, IRW
      COMMON/ BODY2 / NBPFL, NBODYS, NTHET, XB(51), RB(51), ZB(51),
     1          THETA(11), RPP, XBPANL(20)
      COMMON / BODYSP/ DRDX(51), DZDX(51), DRDXC(51)
      COMMON/ NACEL /NACELP, XN(51,2), RN(51,2), ZN(51,2), ARN(2)
      COMMON/CORTRN/ NSYM,NTRANS(20),ISYM(20),VA(3,2,20), BTFM(9,2,20)
      COMMON/DEFAU / BODY, BODYS, BTHET, CPCALC,NCASE, NRUN, DADEG, ARB,
     1          PLANE, POLAR, RP, XNACEL, SYM, THICK,  WING, XMACH,
     2          XNB, XNC, PINF, IPUNCH, SID(20), CKASE, ARW(10), XMOM,
     3          YAW
      COMMON/WANGLE/ ALPHAC(55,10), ALPHAT(55,10), ARWT(55,10)
      COMMON/WINGFM/ XWI(30,20), YWI(30,20), ZWI(30,20),NWPI(20),NCHORD,
     1          NSPAN, ISOLID, NPLANE, ICAMBR, ITWIST,ITHICK,IFORM(10)
     2          , PIVOT(10), DIN(10), NWPANL,XLG(10),XTG(10),BROLL
      COMMON/INPUT / TX(2, 50)
      COMMON/POINTP/TY(4), TYP(4), TYPP(4)
      COMMON/POINTS/ NPTS, NVAR
      DIMENSION RBPANL(11), ZBPANL(11), XC(40), XCR(40), YC(40),
     1         AREA(40)
C
      NBODY = 1
      NBODYS = IFIX( BODYS )
      NTHET = IFIX( BTHET )
C
      CALL BODYPF(XB, RB, ZB, THETA, NBPFL, NTHET,NBODYS, XBS1, XBS2,
     1               DSB1, DSB2, XLAST)
C
      CALL BODYPN(XLAST)
C
C          RESET NC AND NB AFTER MODIFICATION IN BODYPN
C
      PLANE = FLOAT( NPLANE )
      NC = XNC
      NB = XNB
      NC1 = NC - 1
      NB1 = NB - 1
      NBC = NB1*NC1
      NBCT = NBC + 1
      NCOLW = NWING * NB1
      NW2 = 2 * NWING
      THKOMP = 1.0
      WING  =  FLOAT(NWING)
      SOLID =  FLOAT(ISOLID)
      PUNCH =  FLOAT(IPUNCH)
C
      REWIND 11
      WRITE (11,1020) BODYS, BTHET, PLANE, RP, YAW
      WRITE ( 11,1020) ( XB(N), N=1,NBODYS)
      WRITE ( 11,1020) ( RB(N), N=1,NBODYS)
      WRITE ( 11,1020) ( ZB(N), N=1,NBODYS)
      WRITE ( 11,1020) ( THETA(N), N=1,NTHET)
      WRITE (11,1020) XBS1,XBS2,DSB1,DSB2
      WRITE (11,1020) ( XBPANL(N), N=1,NPLANE )
      IF ( RP .GT. 0.0 ) GO TO 610
      WRITE (11,1020) ( RBPANL(N), N=1,NTHET )
      WRITE (11,1020) ( ZBPANL(N), N=1,NTHET )
C
```

```
  410 CONTINUE
      PERCT = 0.1
      DO 450 I = 1, NW2, 2
      NW1 = I/2 + 1
      J = I
      IF ( IFORM(NW1) .EQ. 2 ) J = NW1
      WRITE (11,1020) XWI(1,J),YWI(1,J),XWI(2,J),XWI(4,J),YWI(4,J),
     1           XWI(3,J)
      WRITE (11,1020) XNB, XNC, WCASE, ZWI(1,J)
      WRITE (11,1020) PIVOT(NW1), DIM(NW1), BROLL
      CALL WINGS( NW1, XC, XCR, YC, AREA)
  450 CONTINUE
      TWIST = FLOAT( ITWIST )
      THICK = FLOAT( ITHICK )
      CAMB = FLOAT( ICAMBR )
      WRITE (11,1020) XMACH, SYM, THICK,YAW, WING, BROLL
C
      WRITE (11,1020) CKASE, CPCALC, POLAR, THICK, CAMB,XNACEL, PINF,
     1    PUNCH,SOLID,WING,( SID(K), K=1,20), (IFORM(K), K=1,10),XMOM
C
      IF ( XNACEL .EQ. 0 ) GO TO 460
      CALL NACEL( XNACEL, NWING )
  460 CONTINUE
      WRITE (11,1020) ARB, DADEG
      WRITE (11,1020) (ARW(I), I=1,10), TWIST, NWING
      IF ( ITWIST  .EQ. 1 ) WRITE (11,1020) ( (ARWT(J,I), J=1,NB1), I=1,
     1           NWING )
      IF ( ICAMBR  .EQ. 1 ) WRITE (11,1020) ( ( ALPHAC(I,J), I=1,NBC),
     1           J=1,NWING )
      IF(ITHICK .NE. 1 )  GO TO 480
      DO 470 L = 1, NWING
      DO 470 J = 1, NB1
      KI = 1 + (J-1)*NC
      KK = KI + NC1
      WRITE (11,1020) ( ALPHAT(I,L), I=KI,KK)
  470 CONTINUE
C
      XWTIP = XWI(4, 1)
C
  480 CONTINUE
      REWIND 11
C
 1020 FORMAT (7F10.4)
C
      RETURN
      END
C
C/    LIST,NONE
      SUBROUTINE WINGIN
C
C
      COMMON/ MAIN / LA, LB, LC, LD, LE, LF, LI,LO, NBODY, NWING,XMACHN,
     1          SYMM, KACE, NPOLAR, IRW
      COMMON/DEFAU / BODY, BODYS, BTHET, CPCALC, CASE, NRUN, DADEG, ARB,
     1           PLANE, POLAR, RP, XNACEL, SYM, THICK,  WING, XMACH,
     2           XNB, XNC, PINF, IPUNCH, SID(20), CKASE, ARW(10), XMOM
      COMMON/WANGLE/ ALPHAC(55,10), ALPHAT(55,10), ARWT(55,10)
      COMMON/WINGFM/ XWI(30,20), YWI(30,20), ZWI(30,20),NWPI(20),NCHORD,
     1           NSPAN, ISOLID, NPLANE, ICAMBR, ITWIST,ITHICK,IFORM(10)
     2           , PIVOT(10), DIM(10), NWPANL,XLG(10),XTG(10),BROLL
C
      DIMENSION XC(40), XCR(40), YC(40), AREA(40)
C
C
      NBODY = 0
      THKOMP = 1.0
      NPLANE = IFIX( PLANE )
      PUNCH = FLOAT( IPUNCH )
      SOLID = FLOAT( ISOLID )
      WING  = FLOAT( NWING )
C
      NC = XNC
      NB = XNB
      NC1 = NC - 1
      NB1 = NB - 1
      NBC = NB1*NC1
      NPCT = NBC + 1
      NCOLW = NWING * NB1
C
C        WRITE ON TAPE 11 FOR AERO CODE
C
      REWIND  11
C
      NW2 = 2 * NWING
      DO 50 I=1,NW2,2
      J = I
      NW1 = I / 2 + 1
      IF ( IFORM(NW1) .EQ. 2 ) J = NW1
      WRITE (11,1020) XWI(1,J),YWI(1,J),XWI(2,J),XWI(4,J),YWI(4,J),
     1          XWI(3,J)
      WRITE (11,1020) XNB, XNC, WCASE, ZWI(1,J)
      WRITE (11,1020) PIVOT(NW1), DIN(NW1), BROLL
  50  CONTINUE
C
C        READ IN WING PLAN FORM AND COMPUTE WING CAMBER, TWIST, AND
C        THICKNESS SLPOE.
C
      DO 100 I=1,NWING
      CALL WINGS(I, XC, XCR, YC, AREA)
  100 CONTINUE
      THICK = FLOAT( ITHICK )
      TWIST = FLOAT( ITWIST )
      CAMB = FLOAT( ICAMBR )
      WRITE (11,1020) XMACH, SYM, THICK,YAW, WING, BROLL
C
      WRITE (11,1020) CKASE,CPCALC,POLAR,THICK,CAMB,XNACEL,PINF,
     1     PUNCH,SOLID,WING, ( SID(K), K=1,20), (IFORM(K), K=1,10),XNOM
      WRITE (11,1020) ARB, DADEG
      WRITE (11,1020) (ARW(I), I=1,10), TWIST, NWING
C
      IF ( ITWIST  .EQ. 1 ) WRITE (11,1020) ( (ARWT(J,I), J=1,NB1), I=1,
     1          NWING )
      IF ( ICAMBR  .EQ. 1 ) WRITE (11,1020) ( (ALPHAC(I,J), I=1,NBC),
     1          J=1,NWING )
      IF(ITHICK .NE. 1 )  GO TO 480
      DO 470 L = 1, NWING
      DO 470 J = 1, NB1
      KI = 1 + (J-1)*NC
      KK = KI + NC1
      WRITE (11,1020) ( ALPHAT(I,L), I=KI,KK)
  470 CONTINUE
  480 CONTINUE
C
      REWIND  11
C
 1020 FORMAT ( 7F10.4)
C
C
      RETURN
      END
```

```
C
      SUBROUTINE WINGS( I, XC, XCR, YC, AREA)
C
C
C        COMPUTES WING CAMBER( ALPHAC), THICKNESS SLOPE ( ALPHAT ),
C        TWIST ( ARWT ) AND DIHEDRALS.
C
      COMMON/REPEAT/ IREPT
      COMMON/WANGLE/ ALPHAC(55,10), ALPHAT(55,10), ARWT(55,10)
      COMMON/WINGFM/ XWI(30,20), YWI(30,20), ZWI(30,20),NWPI(20),NCHORD,
     1         NSPAN, ISOLID, NPLANE, ICAMBR, ITWIST,ITHICK,IFORM(10)
     2         , PIVOT(10), DIHED(10), NWPANL, XLG(10), XTG(10)
      DIMENSION XC(40),XCR(40), YC(40), CF(115), ZU(40), ZL(40), ZCM(40)
     1         ,AREA(40)
C
C
      CDR = 57.295779
      NW = 2 * I - 1
      NW1 = NW + 1
      IREPT = 0
      NCH = NCHORD
      NSP = NSPAN
      NC1 = NCH - 1
      NB1 = NSPAN - 1
      NP = ( NCH-1 ) * ( NSP-1 )
      NP1 = NCH + NB1
      NPTU = NWPI(NW)
      NPTL = NWPI(NW1)
      IREPT = 1
C
C        WING SURFACE PANEL CONTROL POINT (XC, YC), AREA (AREA) AND
C      CENTROID (XCR)
C
      CALL WPANEC( XWI(1,NW), YWI(1,NW), NCH, NSP, XL, XCR, YC, AREA)
C
C
C
C        IFORM(I) = 0 , AIRFOIL WING, ZWI FOR UPPER AND LOWER SURFACE
C                      ARE GIVEN AT SAME LOCATIONS ( XWI, YWI )
C                 = 1 ,           ZWI ARE GIVEN AT DIFFERENT LOCALS
C                 = 2 , FLAT WING,
C
      IF ( IFORM(I)  .GT. 1 ) GO TO 450
C
C        DETERMINE THE CAMBER PLANE
      IF ( IFORM(I)  .LT. 1 ) GO TO 100
C
      CALL SURFIT( XWI(1,NW1), YWI(1,NW1),NPTL, ZWI(1,NW1), CF)
      CALL INTERP( NPTL, XWI(1,NW1), YWI(1,NW1), CF, NPTU, XWI(1,NW),
     1         YWI(1,NW), ZWI(1,NW1 ) )
      IREPT = 2
  100 CONTINUE
      DO 200 J = 1, NPTU
      ZCM(J) = 0.5 * ( ZWI(J,NW) + ZWI(J,NW1) )
      ZU(J) = ZWI(J,NW) - ZCM(J)
  200 CONTINUE
C
C        WING  TWIST  ( ARWT )
      CALL WTWIST( YC, I )
C
C
C        WING  CAMBER  ( ALPHAC )
C
C
      CALL SURFIT( XWI(1,NW), YWI(1,NW), NPTU, ZCM, CF)
      CALL INTERP( NPTU, XWI(1,NW),YWI(1,NW),CF,NP,XC,YC,ALPHAC(1,I) )
C
C          WING  THICKNESS SLOPE  ( ALPHAT )
C
      IREPT = 2
      CALL SURFIT( XWI(1,NW), YWI(1, NW), NPTU, ZU, CF)
      CALL INTERP( NPTU, XWI(1,NW),YWI(1,NW),CF,NP,XC,YC,ALPHAT(1,I) )
C
C          SET ITWIST FOR WING
C
      ITWIST = 0
      DO 310 K=1,NB1
      IF ( ARWT(K,I)*CDR  .GT. 0.1 ) GO TO 315
  310 CONTINUE
      GO TO 316
  315 CONTINUE
      ITWIST = 1
  316 CONTINUE
C
C          SET ICAMBR FOR WING
C
      ICAMBR = 0
      DO 410 K=1,NP
      IF ( ALPHAC(K,I)*CDR  .GT. 0.01 ) GO TO 415
  410 CONTINUE
      GO TO 416
  415 CONTINUE
      ICAMBR = 1
  416 CONTINUE
      GO TO 500
C
  450 CONTINUE
C
C          FLAT WING SURFACE
C
      ITWIST = 0
      ICAMBR = 0
      CNX = ( YWI(2, I) - YWI(1,I))*(ZWI(3,I)-ZWI(1,I))-( YWI(3,I)-
     1          YWI(1,I) ) * ( ZWI(2,I) - ZWI(1,I) )
      CNZ = ( XWI(2, I) - XWI(1,I) )*( YWI(3,I)-YWI(1,I) )-( XWI(3,
     1          I) - XWI(1,I) ) *( YWI(2,I) - YWI(1,I) )
      ALP = ATAN( - CNX / CNZ )
      IF ( ALP .LE. 1.0E-3 ) GO TO 600
      DO 470 J =1, NP
      ALPHAT(J,I) = ALP
  470 CONTINUE
C
  500 CONTINUE
C
C          REARRANGE ALPHAT TO CORRESPONDING WITH AERO CODE FORMAT
C
      DO 510 J=1,NP
      CF(J) = ALPHAT(J,I)
  510 CONTINUE
C
C
      DO 520 J=1,NB1
      JNC1 = (J-1)*NC1
      I1 = 1 + JNC1
      J1 = J + JNC1
      ALPHAT(J1,I) = CF(I1)
      DO 515 L = 1, NC1
      IJ = L + JNC1
      IJ1 = IJ + J
      ALPHAT(IJ1,I) = CF(IJ)
  515 CONTINUE
  520 CONTINUE
C
      ITHICK = 1
      GO TO 610
  600 CONTINUE
      ITHICK = 0
  610 CONTINUE
C
      RETURN
      END
C
      SUBROUTINE INTERP( N, X, Y, CF, NP, XP, YP, ZP )
C
C
C        INTERPOLATES AT NP POINTS ( XP, YP )  FOR  DERIVATIVE
C           D/DX
C
      DIMENSION X(1), Y(1), XP(1), YP(1), ZP(1), CF(1)
C
      DO 100 I=1,NP
      SUM = CF(2)
      DO 110 J=1,N
      J3 = J + 3
      RI = (XP(I) - X(J) )**2 + ( YP(I) - Y(J) )**2
      IF ( RI  .LT. 1.E-9 ) GO TO 110
      SUM = SUM+ 2.0*CF(J3)*(1.0+ALOG(RI) )*(XP(I)-X(J) )
  110 CONTINUE
      ZP(I) = SUM
  100 CONTINUE
C
      RETURN
      END
C
      SUBROUTINE WPANEC( XI, YI, NC, NB, XC, XCR, YC, AREA)
C
C
C        COMPUTE CONTROL POINT ( XC, YC ) OF WING PANELS
C
      DIMENSION XI(1), YI(1), XC(1), YC(1), XY(13,11), XW(13), YW(11),
     1          DY(11), XCR(40), AREA(40)
C
      PERCT = 0.1
      XCPT = 0.95
      XW(1) = XI(1)
      XCHORD = XI(2) - XI(1)
      XW(2) = XW(1) + PERCT*XCHORD
      DX = (1.0 - PERCT) * XCHORD / ( NC - 2 )
      DO 100 L=3,NC
      XW(L) = XW(L-1) + DX
  100 CONTINUE
      BY = YI(4) - YI(1)
      DBY = BY / ( NB - 1)
      YW(1) = YI(1)
      DO 165 N=2,NB
      YW(N) = YW(N-1) + DBY
  165 CONTINUE
      SLE = ( XI(4)-XI(1))/ BY
      STE = (XI(3)-XI(2) )/ BY
      IF ( SLE-STE) 210, 230, 210
  210 CONTINUE
      DX = ( XI(3) - XI(4) )/ ( XI(2) - XI(1) )
      DO 215 M=1,NC
      XY( M, NB ) = XI(4) + DX * ( XW(M) - XI( 1 ) )
  215 CONTINUE
      DO 220 N=1,NB
      DY(N) = ( YW(N) - YI(1) ) / BY
      DO 220 M=1,NC
      XY(M, N) = XW(N) + ( XY( M, NB) - XW(N) ) * DY(N)
  220 CONTINUE
      GO TO 250
  230 CONTINUE
      DO 240 N=1,NB
      DY(N) = YW(N) - YI(1)
      DO 240 M=1,NC
      XY( M, N) = XW(N) + SLE* DY(N)
  240 CONTINUE
  250 CONTINUE
C
C        CONTROL POINT COORDINATES ( XC, YC), CENTROID ( XCR ), AND
C       AREA ( AREA )
C
      NC1 = NC-1
      NB1 = NB-1
      DO 350 N=1,NB1
      DO 350 M=1,NC1
      K = M + (N-1)*NC1
      CT = XY(M+1,N+1) - XY(M,N+1)
      CR = XY(M+1,N) - XY(M,N)
      RY = ( 1.0 +CT/(CR+CT) )/3.0
      RY1 = 1.0 - RY
      XWL = RY*XY(M, N+1) + RY1*XY(M, N)
      XWT = RY * XY(M+1, N+1) + RY1*XY(M+1, N)
      XC(K) = XCPT * XWT + ( 1.0 - XCPT) * XWL
      XCR(K) = 0.5*( XWL + XWT)
      YC(K) = YW( N+ 1) * RY + YW(N) * RY1
      YDUM = YW(N+1) - YW(N)
      AREA(K) = 0.5*YDUM*( CR + CT )
  350 CONTINUE
C
      RETURN
      END
C
      SUBROUTINE WTWIST( YC, I )
C
C
C        COMPUTES THE WING TWIST ANGLE  ARWT
C
      COMMON/WANGLE/ ALPHAC(55,10), ALPHAT(55,10), ARWT(55,10)
      COMMON/WINGFM/ XWI(30,20), YWI(30,20), ZWI(30,20),NWPI(20),NCHORD,
     1          NSPAN, ISOLID, NPLANE, ICAMBR, ITWIST,ITHICK,IFORM(10)
     2          , PIVOT(10), DIN(10), NWPANL, XLG(10), XTG(10)
      DIMENSION YC(1), VX(2), VY(2), VZ(2)
C
C     COMPU~ES LEADING AND TRAILING EDGE UNIT VECTORS
C
      DO 100 J=1,2
      J1 = 4
      J2 = 1
      IF ( J .EQ. 2 ) J1 = 3
      IF ( J .EQ. 2 ) J2 = 2
      DX = XWI(J1,I) - XWI(J2,I)
      DY = YWI(J1,I) - YWI(J2,I)
      DZ = ZWI(J1,I) - ZWI(J2,I)
      SUML = SQRT( DX**2 + DY**2 + DZ**2 )
      VX(J) = DX / SUML
      VY(J) = DY / SUML
      VZ(J) = DZ / SUML
  100 CONTINUE
```

```
C
      NB1 = NSPAN - 1
      NC1 = NCHORD - 1
      DO 200 K=1,NB1
      K1 = 1 + ( K-1 ) * NCHORD
      DY = YC( K1 ) - YWI( 1, I )
      XLE = XWI(1,I) + VX(1) * DY / VY(1)
      XTE = XWI(2,I) + VX(2) * DY / VY(2)
      ZLE = ZWI(1,I) + VZ(1) * DY / VY(1)
      ZTE = ZWI(2,I) + VZ(2) * DY / VY(2)
      ARWT(K,I) = ( ZLE-ZTE ) / ( XTE-XLE )
  200 CONTINUE
C
      RETURN
      END
      SUBROUTINE WTRANS( NW, IW )
C
C
C        TRANSFER COORDINATES DEFINING WING FORM INTO THAT OF AERO CODE
C
      COMMON/WINGFM/ XWI(30,20), YWI(30,20), ZWI(30,20),NWPI(20),NCHORD,
     1          NSPAN, ISOLID, NPLANE, ICAMBR, ITWIST,ITHICK,IFORM(10)
     2          , PIVOT(10), DIH(10), NWPANL, XLG(10), XTG(10)
      COMMON/CORTRN/ NSYM, NTRANS(20), ISYM(20), VA(3,2,20),BTFM(9,2,20)
C
      DR = 0.0174532925
      NWP = NWPI( NW )
      IF ( IFORM(IW)  .EQ. 2 ) NWP = 4
      DO 400 J=1,NWP
      R = XWI( J, NW )
      THETA = YWI( J, NW) * DR
      PHI = ZWI( J, NW) * DR
C
      NGO = ISYM( NSYM ) + 1
      GO TO ( 300, 100, 200) , NGO
  100 CONTINUE
      XWI( J, NW) = R * COS( THETA )
      YWI( J, NW) = R * SIN( THETA )
      GO TO 300
C
  200 CONTINUE
      XWI(J,NW) = R * SIN( THETA ) * COS( PHI )
      YWI(J,NW) = R * SIN( THETA ) * SIN( PHI )
      ZWI(J,NW) = R * COS( THETA )
C
  300 CONTINUE
C
  400 CONTINUE
C
      RETURN
      END
C
      SUBROUTINE NACEL( XNACEL, NWING )
C
C
C        INTERPOLATES INPUT OF NACEL CONFIGURATION AND WRITES ON TAPE 11
C        FOR AERO CODE.
C
      COMMON/ BODY2 / NBPFL, NBODYS, NTHET, XB(51), RB(51), ZB(51),
     1          THETA(11), RP,  XBPANL(20)
      COMMON / BODYSP/ DRDX(51), DZDX(51), DRDXC(51)
      COMMON/ NACEL /NACELP, XN(51,2), RN(51,2), ZN(51,2), ARN(2)
      COMMON/CORTRN/ NSYM, NTRANS(20), ISYM(20), VA(3,2,20),BTFM(9,2,20)
C
      ANCEL = ABS( XNACEL )
      NACELS= IFIX(ANCEL)
      NW2 = 2 * NWING
      DO 300 N = 1, NACELS
      CALL BODYPF( XN(1,N), RN(1,N), ZN(1,N), THETA, NACELP, NBODYS, XN1
     1            , XN2, DS1, DS2, XLAST)
      NSYM = 1 + NW2 + N
      YI = VA( 2, 1, NSYM ) + VA( 2, 2, NSYM )
      ZI = VA( 3, 1, NSYM ) + VA( 3, 2, NSYM )
C
      WRITE (11,1020) XN(1,N), YI, ZI, NBODYS, NTHET, ARN(N)
      IF ( XNACEL) 100, 300, 200
C
  100 CONTINUE
      IF ( N - 1 ) 300, 200, 300
  200 CONTINUE
      WRITE (11,1020) ( XN(L,N), L=1,NBODYS )
      WRITE (11,1020) ( RN(L,N), L=1,NBODYS )
      WRITE (11,1020) ( ZN(L,N), L=1,NBODYS )
      WRITE (11,1020) ( THETA(L), L=1,NTHET )
      WRITE (11,1020)  XN1, XN2, DS1, DS2,(DRDX(J),DZDX(J), J=1,NBODYS)
  300 CONTINUE
C
 1020 FORMAT ( 7F10.4 )
C
      RETURN
      END
      SUBROUTINE TRANSM( A, B, C, UNIVEC)
C
C
C              COMPUTES THE TRANSFORM MATRIX TFM BETWEEN TWO COORDINATE
C              SYSTEMS 1 AND 2 . A  DEFINE THE ORIGIN,  B  A POINT ON
C              THE Z AXIS AND C  A POINT ON THE X - Z PLANE OF COORDINA-
C              TE SYSTEM 2.
C         THE MATRIX UNIVEC(L,M) HAS THE ROW INDEX L FOR THE UNIT VECTOR
C           I, J, K AND THE COLUMN INDEX M FOR THEIR THREE COMPONENTS IN
C           THE COORDINATE SYSTEM  1 .
C
      DIMENSION A(3), B(3), C(3), UNIVEC(3,3), UNIVCA( 3 ), N1(3), N2(3)
      DATA N1 / 3, 1, 2 /, N2 / 2, 3, 1 /
C
C         COMPUTE THE UNIT VECTOR  I, J, K OF COORDINATE SYSTEM 2 IN TERM
C           OF COORDINATE SYSTEM 1 .
C
C
C
      SUM = 0.0
      SUM1 = 0.0
      DO 100 I = 1, 3
      SUM = SUM + ( B(I) - A(I) )**2
      SUM1 = SUM1 + ( C(I) - A(I) )**2
  100 CONTINUE
      AB = SQRT ( SUM )
      CA = SQRT( SUM1 )
C
C            UNIT  VECTOR  K
C              UNIT VECTOR  CA
C
      DO 200 J= 1, 3
      UNIVEC(3, J) = ( B(J) - A(J) )/ AB
      UNIVCA( J ) = ( C(J) - A(J) ) / CA
  200 CONTINUE
C
C            UNIT  VECTOR  J
C
      SUM = 0.0
      DO 300 I = 1, 3
      I1 = N1(I)
      I2 = N2(I)
      UNIVEC(2,I) = UNIVCA(I1)*UNIVEC(3,I2)-UNIVCA(I2)*UNIVEC(3,I1)
      SUM = SUM + UNIVEC( 2, I)**2
  300 CONTINUE
      SUM = SQRT( SUM )
      DO 350 I=1,3
      UNIVEC(2,I) = UNIVEC(2,I) / SUM
  350 CONTINUE
C
C            UNIT  VECTOR  I
C
      SUM = 0.0
      DO 400 I = 1, 3
      I1 = N1(I)
      I2 = N2(I)
      UNIVEC( 1, I) = UNIVEC(2, I2) * UNIVEC( 3, I1 ) - UNIVEC( 2, I1 )*
     1         UNIVEC( 3, I2 )
      SUM = SUM + UNIVEC( 1, I)**2
  400 CONTINUE
      SUM = SQRT( SUM )
      DO 450 I=1,3
      UNIVEC(1,I) = UNIVEC(1,I) / SUM
  450 CONTINUE
C
      RETURN
      END
C
      SUBROUTINE CUBERT
      DIMENSION X(3)
      COMMON/LOCTN / V(3), L, M, G(4), VMAX, VMIN
      COMMON/POINTS/ NPTS, NVAR
      L=0
      SUM = SQRT( G(1)**2+G(2)**2+G(3)**2+G(4)**2 )
      G1 = G(1) / SUM
      G2 = G(2) / SUM
      G3 = G(3) / SUM
      G4 = G(4) / SUM
C
C        EQ. 1   G1*X**3 + G2*X**2 + G3*X = 0.
      IF ( ABS( G4) .LT. 1.E-6) GO TO 4
C
C        EQ. 2   G1*X**3 + G2*X**2 + G3*X +G4 = 0.
      IF ( ABS( G1) .GT. 1.E-6) GO TO 9
C
C        EQ. 3   G2*X**2 + G3*X + G4 = 0.
      IF ( ABS( G2) .GT. 1.E-6) GO TO 7
C
C        EQ. 4   G3*X + G4 = 0.
      IF ( ABS( G3) .GT. 1.E-6) GO TO 5
      RETURN
C
  4   CONTINUE
      V(1) = 0.0
      L=1
      RETURN
  5   CONTINUE
      VO    = - G(4) / G(3)
      GO TO 35
  7   CONTINUE
      R1 = G(3)**2 - 4.0*G(2)*G(4)
      IF (  R1  .GT. 0.0 )  GO TO 107
      WRITE (6,901)
  901 FORMAT (1H0,*NAGATIVE RADICAL RESULTED FROM SOLVING THE QUADRATIC
     1EQUATION  G2.X2 + G3.X + G4 = 0*)
      RETURN
  107 CONTINUE
      R1=SQRT(R1)
      V1 = ( -G(3) + R1 ) *0.5 / G(2)
      V(1) = V1
      IF ( V1  .GT. VMIN .AND.  V1  .LT. VMAX ) GO TO 8
      V2 = (-G(3) - R1 )*0.5 / G(2)
      V(2) = V2
      IF ( V2  .GT. VMIN .AND.  V2  .LT. VMAX ) GO TO 8
      WRITE (6,902) VMIN, V1,V2, VMAX
  902 FORMAT (1H0,*BOTH ROOTS V1, V2 OF QUADRATIC EQ. G2.X2 + G3.X + G4
     1= 0 OUTSIDE OF LIMIT VMIN, VMAX*/ 1X,*VMIN =*,E14.4,4X,*V1 =*,
     2           E14.4,4X,*V2 =*,E14.4,4X,*VMAX =*,E14.4)
      RETURN
  8   CONTINUE
      L = 1
      RETURN
C
C        THREE ROOTS
C
  9   CONTINUE
      A1 = G(2) / G(1)
      A2=G(3)/G(1)
      A3=G(4)/G(1)
      A=A2-A1*A1/3.
      Q=2.*A1*A1*A1/27.-A1*A2/3.+A3
      RAD=Q*Q/4.+A*A*A/27.
      SIGN=1.
      IF(Q.GE.0.)SIGN=-1.
      IF(RAD)10,20,20
C
C              THREE UNEQUAL REAL ROOTS
C
  10  CONTINUE
      A3 = SQRT( ABS( A/3.0) )
      A2 = SQRT( ABS( A**3/3. ) )
      PHI = ACOS( ABS( 1.5*Q/ A2) )
      DO 11 I=1,3
      X(I) = 2.0*SIGN*A3*COS((PHI+6.2831853*FLOAT(I-1))/3.0)
  11  CONTINUE
      GO TO 40
C
C        EITHER ONE REAL AND TWO IMAG. OR THREE REAL ROOTS OF WHICH TWO
C        ARE EQUAL
C
  20  CONTINUE
      V1 = 1.0 / 3.0
      Q = A2*V1 - ( A1*V1)**2
      R = .5*A1*A2*V1 - .5 *A3 - ( A1*V1) **3
      RAD = Q**3 + R*R
      IF ( RAD  .GT.  0.0 )  GO TO 120
      WRITE (6,903)
  903 FORMAT (1H0,*ERROR IN TESTING RADICAL, REDUNDENT TEST*)
      RETURN
  120 CONTINUE
      RAD = SQRT( RAD)
      A2 = R + RAD
      SIGN = 1.
      IF ( A2  .LT.  0.0)  SIGN = -1.
      A2 = SIGN * ABS( A2 ) **V1
      A3 = R - RAD
      SIGN = 1.
      IF ( A3 .LT. 0.0 ) SIGN = -1.
      A3 = SIGN * ABS( A3 )**V1
      VO = A2 + A3 - A1*V1
   35 CONTINUE
      IF ( VO .GT. VMIN .AND. VO  .LT. VMAX ) L = 1
      IF ( L  .EQ. 0 ) WRITE (6,905)
  905 FORMAT (1H0,*THE ONLY UNEQUAL ROOT IS OUTSIDE OF VMIN, VMAX*/1X,
     1          *VMIN =*,E14.4,4X,* V =*,E14.4,4X,*VMAX =*,E14.4)
      V(1) = VO
      RETURN
   40 CONTINUE
      DO 45 I=1,3
      VO = X(I) - A1/3.0
      X(I) = VO
      V(I) = VO
      IF ( VO .LT. VMIN .OR.  VO  .GT. VMAX ) GO TO 50
      L=1
      RETURN
   50 CONTINUE
   45 CONTINUE
      WRITE (6,904) VMIN, V(1), V(2), V(3), VMAX
  904 FORMAT (1H0,*ALL THREE UNEQUAL ROOTS OUTSIDE OF LIMIT VMIN, VMAX*/
     1    1X,*VMIN =*,E14.4,4X,*V1 =*,E14.4,4X,*V2 =*,E14.4,4X,*V3 =*,
     2          E14.4,4X,*VMAX =*,E14.4)
C
      RETURN
      END
C
      SUBROUTINE POINT( N )
      COMMON/CURFIT/ B(2,4, 50), T( 50), SCALEF(2)
      COMMON/ ENDPT/ XFIRST, XLAST, XSAVE( 50)
      COMMON/LOCTN/ V(3), L, M, A(4), VMAX, VMIN
      COMMON/POINTS/ NPTS, NVAR
      COMMON/ POINTP/  X(4),  XP( 4),  XPP( 4)
      DIMENSION DX(4)
C
      N1 = NPTS - 1
      IF(N.NE.0)GO TO 4
      DO 1 I=1,N1
      M=I
      IF ( V(1)  .LE. T(I+1) ) GO TO 2
    1 CONTINUE
      WRITE(6,100)
  100 FORMAT(1H0,33H ERROR - PARAMETER EXCEEDS LIMITS)
      RETURN
    2 CONTINUE
      DO 300 K=1,3
      P = V(K)
      DX(K) = 1.E10
      IF ( P .LT. VMIN  .OR.  P  .GT. VMAX ) GO TO 300
      DX(K) = ABS((((B(1,1,M)*P + B(1,2,M))*P + B(1,3,M))*P + B(1,4,M))*
     1          SCALEF(1) - XIND )
  300 CONTINUE
      IF ( DX(1)  .GT. DX(2) ) GO TO 302
      IDENT = 1
      IF ( DX(1)  .GT. DX(3) )  IDENT = 3
      GO TO 305
  302 CONTINUE
      IDENT = 2
      IF ( DX(2)  .GT. DX(3) )  IDENT = 3
  305 CONTINUE
      P = V(IDENT)
C
      DO 3   I=1, NVAR
      X(I)=(((B(I,1,M)*P+B(I,2,M))*P+B(I,3,M))*P+B(I,4,M))*SCALEF(I)
      XP(I)=((3.*B(I,1,M)*P+2.*B(I,2,M))*P+B(I,3,M))*SCALEF(I)
    3 XPP(I)=(6.*B(I,1,M)*P+2.*B(I,2,M))*SCALEF(I)
```

```
C
  310 CONTINUE
      RETURN
C
    4 X(N)=X(N)/SCALEF(N)
      DO 5 I=1,N1
      M=I
      K = I + 1
      IF ( XSAVE(K)  .GE. X(N) ) GO TO 6
    5 CONTINUE
      XIND = X(N) * SCALEF(N)
      XFT = XFIRST * SCALEF(1)
      XLT = XLAST * SCALEF(1)
      WRITE (6,101)  XIND, XFT, XLT
  101 FORMAT (1H0,*  THE GIVEN VALUE OF INDEPENDENT VARIABLE *,E12.4,
     1           5X,*IS OUT OF RANGE TO BE INTERPOLATED  *,E12.4,5X,E12.4)
      RETURN
    6 DO 7 I=1,4
    7 A(I)=B(N,I,M)
      A(4)=A(4)-X(N)
      IF ( XFIRST  .EQ. X(N) ) GO TO 112
      IF ( XLAST   .EQ. X(1) ) GO TO 111
      VMAX = T(M+1) * 1.0001
      VMIN = T(M) * 0.9999
      CALL CUBERT
      XIND = X(1)* SCALEF(1)
      GO TO 103
  112 V(1) = 0.0
      L=1
      GO TO 103
  111 V(1) = 1.0
      L=1
  103 CONTINUE
      IF(L.NE.0)GO TO 2
      OP = P
      WRITE (6,10) OP
   10 FORMAT (1X4HOP =,E15.8)
      DO 8 J=1,2
      K = M + J - 1
      B1 = B(N,1,K)
      B2 = B(N,2,K)
      B3 = B(N,3,K)
      B4 = B(N,4,K)
      P = T(K)
      SUM = B1*P**3 + B2*P**2 + B3*P + B4
      DIF = SUM - X(N)
    8 WRITE (6,9) K,P,B1,B2,B3,B4,X(N),SUM,DIF
    9 FORMAT (1XI3,8E15.8)
      WRITE(6,102)
  102 FORMAT(1H0,32H FAILURE IN CUBE ROOT EXTRACTION)
      RETURN
      END
C
      SUBROUTINE SPLFIT
      REAL L,M,MU
      COMMON/CURFIT/ B(2,4, 50), T( 50), SCALEF(2)
      COMMON/ ENDPT/ XFIRST, XLAST, XSAVE( 50)
      COMMON/INPUT /  X( 2,  50)
      COMMON/POINTS/ NPTS, NVAR
      DIMENSION M(2, 50), S( 50), L( 50), MU( 50), D( 50), P( 50), Q( 50
     1          ), U( 50)
      IF ( NPTS .GT.  50) NPTS= 50
      DO 200 I=1, NVAR
  200 SCALEF(I)=0.
      DO 201 I=1,NPTS
```

AD-A081 596   MCDONNELL DOUGLAS ASTRONAUTICS CO HUNTINGTON BEACH CA   F/G 9/2
AERODYNAMIC COMPUTER CODE FOR COMPUTING PRESSURE LOADS ON A MIS--ETC(U)
DEC 79   K K WANG   N00014-75-C-0462
UNCLASSIFIED   MDC-G8223   NL

2 OF 3
AD
A081596

```
      DO 201 J=1,NVAR
  201 SCALEF(J)=AMAX1(SCALEF(J),ABS(X(J,I)))
      DO 101 J=1,NVAR
      IF ( SCALEF(J)  .LT. 1.E-10 ) SCALEF(J) = 1.0
  101 CONTINUE
      DO 202 I=1,NPTS
      DO 202 J=1,NVAR
  202 X(J,I)=X(J,I)/SCALEF(J)
      DO 203 I=1,NPTS
      XSAVE(I) = X(1,I)
  203  CONTINUE
      XFIRST = X(1,1)
      XLAST = X(1,NPTS)
      T(1)=0.
      S(1)=0.
      SUM=0.
      L(1)=-2.
      MU(1)=0.
      L(NPTS)=0.
      MU(NPTS)=-2.
      D(1)=0.
      D(NPTS)=0.
      P(1)=2.
      Q(1)=1.
      U(1)=0.
      N1=NPTS-1
      DO 2 I=2,NPTS
      SUM1=0.
      DO 1 J=1,NVAR
    1 SUM1=SUM1+(X(J,I)-X(J,I-1))**2
      S(I)=SQRT(SUM1)
    2 T(I)=T(I-1)+S(I)
      DO 3 I=2,NPTS
      S(I)=S(I)/T(NPTS)
    3 T(I)=T(I)/T(NPTS)
      DO 4 I=2,N1
      L(I)=S(I+1)/(S(I)+S(I+1))
      MU(I)=1.-L(I)
      P(I)=MU(I)*Q(I-1)+2.
    4 Q(I)=-L(I)/P(I)
      P(NPTS)=2.+(1.-Q(N1))
      DO 7 I=1,NVAR
      DO 5 J=2,N1
      D(J)=6.*((X(I,J+1)-X(I,J))/S(J+1)-(X(I,J)-X(I,J-1))/S(J))/(S(J)+
     1S(J+1))
    5 U(J)=(D(J)-MU(J)*U(J-1))/P(J)
      U(NPTS)=-MU(NPTS)*U(N1)/P(NPTS)
      M(I,NPTS)=U(NPTS)
      DO 6 J=1,N1
      K=NPTS-J
      M(I,K)=Q(K)*M(I,K +1)+U(K)
      B(I,1,K)=(M(I,K +1)-M(I,K))/(6.*S(K+1))
      B(I,2,K)=.5*(M(I,K)*T(K+1)-M(I,K+1)*T(K))/S(K+1)
      B(I,3,K)=(.5*(M(I,K+1)*T(K)**2-M(I,K)*T(K+1)**2)+X(I,K+1)-X(I,K)+
     1(M(I,K)-M(I,K+1))*S(K+1)**2/6.)/S(K+1)
    6 B(I,4,K)=(M(I,K)*T(K+1)**3-M(I,K+1)*T(K)**3)/(6.*S(K+1))+(X(I,K)*T
     1(K+1)-X(I,K+1)*T(K))/S(K+1)+S(K+1)/6.*(M(I,K+1)*T(K)-M(I,K)*T(K+1)
     1)
    7 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE SURFIT( X, Y, N, B, TN)
C
```

```
C        COMPUTES COEFFICIENTS FOR SURFACE SPLINE.
C        N  TOTAL NUMBER OF INPUT POINTS, CAN BE RANDOMLY LOCATED.
C        JEQ  FOR EQUATION NUMBERING
C        MUK  FOR COEFFICIENT NUMBERING
C
C        CF  ARE THE COEFFICINT MATRIX OF EQUATION
C              W(X,Y) = B(1) + B(2)*X + B(3)*Y + SUM OF B(I) * R(I)**2 *
C              LN( R(I)**2)
C
      COMMON/REPEAT/ IREPT
      DIMENSION IPS(115), CF(115,115), TM(1), SCALES(115)
      DIMENSION X( 1), Y( 1), B( 1)
C
      N4 = 115
      N1 = N + 1
      N2 = N + 2
      N3 = N + 3
C
C        BYPASS MATRIX INVERSION FOR THE CASE WHERE THE DATA POINTS (X,
C        Y )  ARE IDENTICAL FOR INTERPOLATION
C
C
      IF ( IREPT .GT. 1 ) GO TO 300
      DO 100 JEQ = 1, N
      CF( JEQ, 1) = 1.0
      CF ( JEQ, 2 ) =   X( JEQ )
      CF ( JEQ, 3 ) =   Y( JEQ )
      DO 90 MUK = 1, N
      MUK1 = MUK + 3
      IF ( MUK  .EQ.  JEQ ) GO TO 85
      RJM = ( X( MUK ) - X( JEQ ) )**2 + ( Y( MUK ) - Y( JEQ ) )**2
      CF( JEQ, MUK1 ) = RJM * ALOG( RJM )
      GO TO 90
   85 CF( JEQ, MUK1 ) = 0.0
   90 CONTINUE
  100 CONTINUE
C
      DO 150 K=1,3
      CF( N1, K) = 0.0
      CF( N2, K) = 0.0
      CF( N3, K) = 0.0
  150 CONTINUE
      DO 200  MUK =1,N
      MUK1 = MUK + 3
      CF( N1, MUK1) = 1.0
      CF ( N2, MUK1) =   X( MUK )
      CF( N3, MUK1 ) =   Y( MUK )
  200 CONTINUE
C
C
C
  250 CONTINUE
C
      CALL DECOMP( N3, CF, IPS, SCALES)
C
  300 CONTINUE
C
      B(N1) = B(N2) = B(N3) = 0.0
      CALL SOLVE( N3, CF, B, TM,IPS)
C
C
      RETURN
      END
      SUBROUTINE DECOMP( NN, A, IPS, SCALES)
C
```

```
C
C        DECOMPOSING OF MATRIX A
C
      DIMENSION A(115, 115), SCALES( 1), IPS( 1)
C
      N = NN
C
C        INITIALIZE IPS, A AND SCALES
C
      DO 5 I = 1, N
      IPS(I) = I
      ROWNRM = 0.0
      DO 2 J = 1, N
      IF ( ROWNRM - ABS( A( I,J) ) ) 1, 2, 2
    1 ROWNRM = ABS( A( I, J) )
    2 CONTINUE
      IF ( ROWNRM ) 3, 4, 3
    3 SCALES(I) = 1.0 / ROWNRM
      GO TO 5
    4 CALL SING(1)
      SCALES(I) = 0.0
    5 CONTINUE
C
C        GAUSSIAN ELIMINATION WITH PARTIAL PIVOTING
C
      NM1 = N - 1
      DO 17 K = 1, NM1
      BIG = 0.0
      DO 11 I = K, N
      IP = IPS(I)
      SIZE = ABS( A( IP, K) ) * SCALES(IP)
      IF ( SIZE - BIG ) 11, 11, 10
   10 BIG = SIZE
      IDXPIV = I
   11 CONTINUE
      IF ( BIG) 13, 12, 13
   12 CALL SING(2)
      GO TO 17
   13 IF ( IDXPIV - K) 14, 15, 14
   14 J = IPS(K)
      IPS(K) = IPS( IDXPIV )
      IPS(IDXPIV) = J
   15 KP = IPS(K)
      PIVOT = A( KP, K)
      KP1 = K + 1
      DO 16 I = KP1, N
      IP = IPS(I)
      EM = - A( IP, K) / PIVOT
      A( IP, K) = - EM
      DO 16 J = KP1, N
      A( IP, J) = A( IP, J ) + EM * A( KP, J )
C
C        INNER LOOP
C
   16 CONTINUE
   17 CONTINUE
      KP = IPS(N)
      IF ( A( KP, N ) ) 19, 18, 19
   18 CALL SING(2)
   19 CONTINUE
C
      RETURN
      END
      SUBROUTINE SOLVE( NN, A, B, X, IPS)
C
```

```
C        SOLVE FOR MATRIX  A  *  X  =  B
C
C
      DIMENSION A(115, 115), B( 1), X( 1), IPS(1)
C
      N = NN
      NP1 = N + 1
C
      IP = IPS(1)
      X(1) = B(IP)
      DO 2 I=2,N
      IP = IPS(I)
      IM1 = I - 1
      SUM = 0.0
      DO 1 J = 1, IM1
    1 SUM = SUM + A(IP, J) * X(J)
    2 X(I) = B(IP) - SUM
C
      IP = IPS(N)
      X(N)=X(N) / A( IP, N)
      DO 4 IBACK = 2, N
C
C        I  GOES FROM  (N-1) . . . . . 1
C
      I = NP1 - IBACK
      IP = IPS(I)
      IP1 = I + 1
      SUM = 0.0
      DO 3 J = IP1, N
    3 SUM = SUM + A( IP, J) * X(J)
    4 X(I)=( X(I) - SUM ) / A( IP, I)
C
C
      RETURN
      END
      SUBROUTINE SING(IWHY)
C
C        PRINTS STATEMENT
C
C        NOUT = STANDARD OUTPUT UNIT
C
      NOUT = 6
      GO TO ( 1, 2, 3), IWHY
    1 WRITE ( NOUT, 11)
      GO TO 10
    2 WRITE ( NOUT, 12)
      GO TO 10
    3 WRITE ( NOUT, 13)
C
   11 FORMAT (1HO,*MATRIX WITH ZERO ROW IN DECOMPOSE*)
   12 FORMAT (1HO,*SINGULAR MATRIX IN DECOMPOSE, ZERO DIVIDE IN SOLVE*)
   13 FORMAT (1HO,*NO CONVERGENCE IN IMPRUV. MATRIX NEARLY SINGULAR*)
C
   10 CONTINUE
C
      RETURN
      END
      OVERLAY(WANG, 2, 1)
      PROGRAM HIGHSP
C
C
      COMMON/MAIN /LA, LB, LC, LD, LE, LF, LI,LO, NBODY, NWING, XMACHN,
     1        SYMM, KACE, NPOLAR, IRW
      COMMON/ BODY2 / NBPFL, NBODYS, NTHET, XB(51), RB(51), ZB(51),
     1        THETA(11), RPP, XBPANL(20)
```

```
      COMMON / BODYSP/ DRDX(51), D2DX(51), DRDXC(51),XWROOT,XWTIP
      COMMON/CORTRN/ NSYM,NTRANS(20),ISYM(20),VA(3,2,20), BTFM(9,2,20)
      COMMON/DEFAU / BODY, BODYS, BTHET, CPCALC,WCASE, NRUN, DADEG, ARB,
     1         PLANE, POLAR, RP, XNACEL, SYM, THICK, WING, XMACH,
     2         XNB, XNC, PINF, IPUNCH, SID(20), CKASE, ARW(10), XMOM,
     3         YAW
      COMMON/WANGLE/ ALPHAC(55,10), ALPHAT(55,10), ARWT(55,10)
      COMMON/WINGFM/ XWI(30,20), YWI(30,20), ZWI(30,20),NWPI(20),NCHORD,
     1         NSPAN, ISOLID, NPLANE, ICAMBR, ITWIST,ITHICK,IFORM(10)
     2         , PIVOT(10), DIN(10), NWPANL,XLG(10),XTG(10),BROLL
      COMMON/INPUT / TX(2, 50)
      COMMON/POINTP/TY(4), TYP(4), TYPP(4)
      COMMON/POINTS/ NPTS, NVAR
      DIMENSION XC(40), XCR(40), YC(40), AREA(40)
C
      RD = 57.29577957
      NBODY = 1
      NBODYS = IFIX( BODYS )
      NTHET = IFIX( BTHET )
      NC =XNC
      NB = XNB
      NC1 = NC - 1
      NB1 = NB - 1
      NBC = NB1*NC1
      NW2 = 2*NWING
C
      CALL BODYPF(XB, RB, ZB, THETA, NBPFL, NTHET,NBODYS, XBS1, XBS2,
     1             DSB1, DSB2, XLAST)
C
C          SET DRDX(1) = SLOPE OF THE TIP OF BODY
C
      DO 90 K=1, NBODYS
      DRDX(NBODYS+2-K) = DRDX(NBODYS+1-K)
   90 CONTINUE
      DRDX(1) = DSB1
      PERCT = 0.1
      TWIST = FLOAT( ITWIST )
      THICK = FLOAT( ITHICK )
      CAMB = FLOAT( ICAMBR )
      WING = FLOAT(NWING)
      SOLID = FLOAT(ISOLID)
      PUNCH = FLOAT(IPUNCH)
      YAW = YAW * RD
      BROLL = BROLL * RD
C
      REWIND 11
      WRITE (11,800) XMACH, PINF, POLAR, PUNCH, THICK, CAMB, TWIST,
     1               SOLID, WING, XNC, XNB, BODYS, BTHET,(SID(K), K=1,
     2               20), (IFORM(K), K=1,10), XMOM, ARB, YAW, BROLL
C
      WRITE (11,800) (XB(K), K=1,NBODYS)
      WRITE (11,800) (RB(K), K=1,NBODYS)
      WRITE (11,800) (THETA(K), K=1,NTHET)
C
      DO 400 I = 1, NW2, 2
      NW1 = I/2 + 1
      J = I
      IF ( IFORM(NW1) .EQ. 2 ) J = NW1
C
      CALL WINGS( NW1, XC, XCR, YC, AREA)
      WRITE (11,800) (XC(K), K=1,NBC)
      WRITE (11,800) (XCR(K), K=1,NBC)
      WRITE (11,800) (YC(K), K=1,NBC)
      WRITE (11,800) (AREA(K), K=1, NBC)
      WRITE (11,800) (ARWT(K, NW1), K=1, NB1)
      WRITE (11,800) (ALPHAC(K, NW1), K=1,NBC)
      DO 100 L=1,NB1
      KS = 1 + (L-1)*NC
      KE = KS + NC1
      WRITE (11,800) (ALPHAT(K, NW1), K=KS, KE)
  100 CONTINUE
C
  400 CONTINUE
C
      REWIND 11
C
C          SET WING ROOT (XWROOT) AND WING TIP (XWTIP) VALUES
C
      XWROOT = XLG(1)
      XWTIP = XWI(4, 1)
C
  800 FORMAT (7F10.4)
C
C
      END
      OVERLAY(WOOD, 3, 0)
      PROGRAM AMATE
C
C
C        CONTROL ROUTINE FOR COMPUTING AERODYNAMIC INFLUENCE COEFF.
C       MATRIX AND VELOCITIES
C
      COMMON/MAIN/ NTAPEA,NTAPEB,NTAPEC,NTAPED,NTAPEE,NTAPEF,NTAPEI,
     1             NTAPEO,NBODY,NWING,XMACH,SYM,KACE,NPOLAR,IRW
      COMMON/BLOCK/ALPHAS(210),AREA(210),A(210),ALPHAC(110),ALPHAT(110)
     1            ,CHORD(210),ISYM,NPART(210),NPANEL,NROW(2),NWS,THETA(
     2             210),TAIL,THKOMP,YAW,BROLL,U(210),V(210),VPM(210),
     3             VV(210),VPMM(210),W(210),WPM(210),WW(210),WPMM(210),
     4             X(210,3,4),XBAR(210),XC(210),Y(210,3,4),YBAR(210),
     5             YC(210),Z(210,3,4),ZBAR(210),ZC(210)
C
C
C     READ IN GEOMETRICAL DATA
      CALL INTAPE
C
C     KACE = 1      WING ALONE
C     KACE = 2      BODY ALONE
C     KACE = 3      WING-BODY COMBINATION
C
      GO TO (1000,1030,1000),KACE
C
C     COMPUTE VELOCITY COMPONENTS DUE TO WING SOURCES
C     THICKNESS CASE
 1000 IF (THKOMP .EQ. 0.0) GO TO 1010
      NS=NBODY+1
      CALL EVAL(NS,XBAR,1.0)
C
      GO TO 1020
 1010 CALL FOREOF (NTAPEA)
      CALL FOREOF (NTAPEB)
C     COMPUTE VELOCITY COMPONENTS TO WING PANEL SINGULARITIES
C     LIFTING CASE
 1020 NS=1
      CALL EVAL(NS,XC,0.0)
C
 1030 REWIND NTAPEC
      REWIND NTAPEA
      REWIND NTAPEB
C
C -----------------
```

```
C -----------------
C
C
      END
C
      SUBROUTINE EVAL(NI,XCPT,THKW)
C
C .....................................................................
C     COMPUTES AERODYNAMIC INFLUENCE COEFFICIENTS MATRIX AND VELOCITY
C     COMPONENTS MATRIX
C .....................................................................
C
C
      COMMON/ MAIN / NTAPEA,NTAPEB,NTAPEC,NTAPED,NTAPEE,NTAPEF,NTAPEI,
     1          NTAPEO,NBODY,NWING,XMACH,SYM,KACE,NPOLAR,IRW
      COMMON /PANVEL/ VS1(110),VS2(110),WSS(110)
      COMMON /BLOCK/ALPHAS(210),AREA(210),A(210),ALPHAC(110),ALPHAT(110)
     1,CHORD(210)
     2,ISYM
     3,NPART(210),NPANEL,NROW(2), NWS
     4,THETA(210),TAIL,THKOMP,YAW, BROLL
     5,U(210)
     6,V(210),VPM(210),VV(210),VPMM(210)
     7,W(210),WPM(210),WW(210),WPMM(210)
     8,X(210,3,4),XBAR(210),XC(210)
     9,Y(210,3,4),YBAR(210),YC(210)
     A,Z(210,3,4),ZBAR(210),ZC(210)
C
      DIMENSION AS(210),BBETAN(4),C(4),CPM(4),CPMM(4),Q(4)
     1,R(4),RPM(4),RPMM(4),UU(210),US(210),VS(210),WS(210)
     2,XCPT(210)
C
      ASINH(B)=ALOG(B+SQRT(1.+B*B))
      ACOSH(E)=2.*ALOG(SQRT(ABS((E+1.)/2.))+SQRT(ABS((E-1.)/2.)))
C
C
      EPS=0.0001
      BETA=SQRT(ABS(XMACH**2-1.))
      PI=3.14159265358979
      CONSTA=BETA/4.0
      CONSTB=CONSTA/PI
      CONSTD=BETA/PI
      LYSD=THKW
      IT=1
C
C     INFLUENCING PANEL I
C
      GO TO (1000,2580,1010),KACE
 1000 NC=NWING/NROW(1)
      NR=NROW(1)
      I1=1
      I2=NWING
      GO TO 1050
 1010 IF(THKW)1020,1040,1020
 1020 NC=NWING/NROW(2)
      NR=NROW(2)
      NCT=NI-1
      I1=NI
      I2=NPANEL
      DO 1030 N=1,NBODY
 1030 XCPT(N)=XC(N)
      GO TO 1060
 1040 NC=NBODY/NROW(1)
      NR=NROW(1)
      I1=1
      I2=NBODY
 1050 NCT=0
 1060 N=1
C
C
C *****************
      DO 2530 I=I1,I2
C ******************
 1080 IF(LVSD)1090,1130,1090
 1090 SL=1.0
      IT=1+I-NI-NR*(N-1)
      IF(IT)1110,1110,1100
 1100 N=N+1
      LVSD=1
      GO TO 1140
 1110 LVSD=-1
      FCT=NCT+1
      GO TO 1140
 1120 SL=-1.0
      LVSD=1
      GO TO 1140
 1130 NCT=NCT+1
 1140 COST=COS(THETA(I))
      SINT=SIN(THETA(I))
      TANA= TAN(ALPHAS(I))
C
C
      ABR=BETA*TANA
      IF(ABR-1.0)1150,2570,2570
 1150 IF(ABS(ABR)-EPS)1160,1160,1170
 1160 ABR=0.0
      SA=1.0
      GO TO 1180
 1170 SA=SIGN(1.,ABR)
 1180 APM2=ABR*ABR
      IN=I-NBODY
      NPT=NPART(I)
      TEST=ABS(Y(I,1,1))+ABS(ABS(THETA(I))-.5*PI)
C
C         BYPASS SYMMETRY TEST FOR VERTICAL FIN BY SETTINGT TEST=1.0
C
C
C
C     INFLUENCED PANEL J
C
      DO 2430 J=1,NPANEL
      JN=J-NBODY
      U(J)=0.0
      V(J)=0.0
      W(J)=0.0
      VPM(J)=0.
      WPM(J)=0.
      VPMM(J)=0.
      WPMM(J)=0.
      SINA=SIN(ALPHAS(J))
      COSA=COS(ALPHAS(J))
C
      DO 2430M=1,NPT
      BBETAM(1)=(X(I,M,2)-X(I,M,1))/((Y(I,M,2)-Y(I,M,1))*COST
     1+(Z(I,M,2)-Z(I,M,1))*SINT)
      BBETAM(3)=(X(I,M,4)-X(I,M,3))/((Y(I,M,4)-Y(I,M,3))*COST
     1+(Z(I,M,4)-Z(I,M,3))*SINT)
      BPM1=BBETAM(1)
      BPM3=BBETAM(3)
      IF(BETA)1190,1200,1190
 1190 BBETAM(1)=BBETAM(1)/BETA
      BBETAM(3)=BBETAM(3)/BETA
 1200 BBETAM(2)=BBETAM(1)
      BBETAM(4)=BBETAM(3)
C
C     CORNER POINTS K OF INFLUENCING PANEL I
C
      DO 2420K=1,4
C
      R(K)=0.0
      C(K)=0.0
      Q(K)=0.0
      RPM(K)=0.
      CPM(K)=0.
      RPMM(K)=0.
      CPMM(K)=0.
      BPM=BBETAM(K)
      IF(LVSD)1240,1240,1210
 1210 IF(SL)1220,1240,1230
 1220 IF(K-2)2420,2420,1240
 1230 IF(K-2)1240,1240,2420
 1240 IF(ABS(BPM)-EPS)1260,1260,1250
 1250 SM=SIGN(1.0,BPM)
      GO TO 1270
 1260 SM=1.0
      BPM=0.0
 1270 IF (ABS(ABS(BPM)-1.0)-EPS)1280,1280,1290
 1280 BPM=SM
 1290 EPM=SM*BPM
      BPM2=BPM*BPM
      CONSTC=1.+APM2*BPM2
C
C     CONFIGURATION SYMMETRY CONDITION
C
C
      DO 2410 ISIDE=1,ISYM
      IF(ISIDE.EQ.2.AND.TEST.LE.EPS) GO TO 2410
      SN=(-1.0)**ISIDE
      DELTAT = THETA(I) + THETA(J)*SN
      IF(ABS(DELTAT).LE.EPS) DELTAT=0.
      SINDT=SIN(DELTAT)
      COSDT=COS(DELTAT)
      DELTAY=SN*(-YC(J))-Y(I,M,K)
      DELTAZ= ZC(J)-Z(I,M,K)
C     IF ( KACE .EQ. 1 ) GO TO 200
C     IF ( I .LE. NBODY .AND. J .LE. NBODY ) GO TO 200
C     DELTAZ = -SN*ZC(J) - Z(I,M,K)
C     IF ( BROLL .GT. 0.01) SR=(-1.0)**(ISIDE
C    1            + 1 )
C 200 CONTINUE
      XPM=XCPT(J)-X(I,M,K)
      YPM=(DELTAY*COST+DELTAZ*SINT)*SM
      ZBR=DELTAZ*COST-DELTAY*SINT
      IF (ABS(ZBR)-EPS)1300,1300,1310
 1300 ZBR=0.0
 1310 SZ=SIGN(1.,ZBR)
      ZPM2=ZBR*ZBR
      IF (ABS(YPM)-EPS)1320,1320,1330
 1320 YPM=0.0
 1330 SY=SIGN(1.,YPM)
      XIPM=XPM/BETA
      XIPM2=XIPM*XIPM
      YPM2=YPM*YPM
 1340 CONTINUE
C
```

```
C     VELOCITY COMPONENTS INDUCED BY PANELS PARALLEL TO X- AXIS
      ZPM=ZBR
      TERMAA=XIPM-BPM*YPM
      TERMA=TERMAA**2
      TERMB=YPM2+ZPM2
      TERMC=SQRT(TERMB)
      IF(XMACH-1.)1580,1810,1350
 1350 TERMD=BPM2-1.
      TERME=-TERMD
      TERMG=TERMA+TERMD*ZPM2
      IF(XIPM-TERMC)1360,1530,1370
 1360 IF(BPM2-1.)1510,1570,1570
 1370 IF(ZPM)1380,1410,1380
 1380 FTR1=(BPM*TERMB-XIPM*YPM)/SQRT(TERMB*TERMG)
      IF(FTR1+1.)1650,1650,1390
 1390 IF(FTR1-1.)1400,1420,1420
 1400 FTR1=ACOS(FTR1)*SZ
      GO TO 1460
 1410 IF(YPM)1420,1440,1430
 1420 FTR1=0.
      GO TO 1460
 1430 IF(XIPM-BPM*YPM)1420,1440,1450
 1440 FTR1=PI/2.
      GO TO 1460
 1450 FTR1=PI*SZ
 1460 IF(BPM2-1.)1470,1480,1480
 1470 FTR2=-SQRT(TERME)*ACOS((BPM*XIPM-YPM)/SQRT(TERMG))
      GO TO 1490
 1480 FTR2=SQRT(TERMD)*ACOSH((BPM*XIPM-YPM)/SQRT(TERMG))
 1490 FTR4=SQRT(XIPM2-TERMB)/TERMB
      FTR5=ACOSH(XIPM/TERMC)
      D=CONSTB*(BPM*FTR5+YPM*FTR4-FTR2)
      S=CONSTB*(BPM*FTR1-ZPM*FTR4)
      P=FTR1/(4.*PI)
      GO TO 2370
 1510 IF(YPM)1570,1570,1520
 1520 IF (XIPM-(BPM*YPM+SQRT(TERME)*ABS(ZPM)))1570,1540,1530
 1530 IF (YPM-BPM*XIPM)1570,1540,1550
 1540 CT=.5
      GO TO 1560
 1550. CT=1.0
 1560 D=CONSTA*SQRT(TERME)*CT
      S=CONSTA*BPM*SZ*CT
      P=CT*SZ/4.
      GO TO 2370
 1570 D=0.0
      S=0.0
      P=0.0
      GO TO 2370
 1580 TERMD=1.-BPM2
      TERME=SQRT(TERMD)
      IF(ZPM)1680,1590,1680
 1590 IF(TERMAA*ST)1600,1610,1620
 1600 FTR1=0.
      GO TO 1630
 1610 FTR1=PI/2.
      GO TO 1630
 1620 FTR1=PI
 1630 IF(YPM)1640,1720,1720
 1640 TERMG=TERMAA/(YPM*BPM*XIPM)
      IF(ABS(TERMG)-.001)1650,1650,1720
 1650 TERMN=TERMAA/ABS(YPM)
      IF(TERMN)1660,1670,1660
 1660 FTR2=ALOG(TERMAA*TERMN/(2.*TERMD)*(1.+BPM*TERMN/TERMD))
      GO TO 1730
```

```
1670 FTR2=ALOG(TERMD)
     GO TO 1730
1680 FTR1=(BPM*TERMB-XIPM*YPM)/SQRT(TERMB*(TERMA+TERMD*ZPM2))
     IF(FTR1+1.)1740,1740,1690
1690 IF(FTR1-1.)1710,1700,1700
1700 FTR1=0.
     GO TO 1720
1710 FTR1=ACOS(FTR1)*SZ
1720 FTR2= YPM+BPM*XIPM+TERME*SQRT(XIPM2+TERMB)
     IF (FTR2 .LE. 0.0) GO TO 1650
     FTR2 = ALOG(FTR2)
1730 FTR3=(XIPM+SQRT(XIPM2+TERMB))/TERMB
     IF(FTR3)1760,1750,1760
1740 FTR1 = PI*SZ
     GO TO 1720
1750 FTR4=0.
     GO TO 1770
1760 FTR4=BPM*ALOG(FTR3)
1770 IF(ZPM)1780,1790,1780
1780 FTR5=ATAN(YPM/ZPM)
     GO TO 1800
1790 FTR5=PI/2.*SY
1800 P=(FTR1+FTR5)/(8.*PI)
     S=CONSTB*(BPM*(FTR1+FTR5)-ZPM*FTR3)/2.
     D=CONSTB*(YPM*FTR3+FTR4-TERME*FTR2)/2.
     GO TO 2370
1810 IF(XPM)1570,1570,1820
1820 TERMD=BPM**2
     TERMA=(XPM-BPM*YPM)**2
     IF(ZPM)1830,1840,1830
1830 FTR1=ACOS((-XPM*YPM+BPM*TERMB)/SQRT(TERMB*(TERMA+TERMD*ZPM2)))*SZ
     GO TO 1880
1840 IF(YPM)1850,1850,1860
1850 FTR1=0.
     GO TO 1880
1860 IF(XPM-BPM*YPM)1850,1850,1870
1870 FTR1=PI
1880 P=FTR1/(4.*PI)
     S=-(ABS(XPM)*ZPM/TERMB-BPM*FTR1)/(4.*PI)
     IF(BPM)1890,1900,1890
1890 D=(ABS(XPM)*YPM/TERMB-BPM*ALOG(BPM*TERMC/SQRT(TERMA+ZPM2*TERMD)))
    1/(4.*PI)
     GO TO 2370
1900 D=ABS(XPM)*YPM/(TERMB*PI*4.)
     GO TO 2370
1910 CONTINUE
C
C    VELOCITY COMPONENTS INDUCED BY WING SOURCES
     ZPM=ZER
     TERMAA=XIPM-BPM*YPM
     TERMA=(XIPM-BPM*YPM)**2
     TERMB=YPM2+ZPM2
     TERMC=SQRT(TERMB)
     IF(XMACH-1.)2190,2370,1920
1920 TERMD=BPM2-1.
     TERME=-TERMD
     IF (XIPM-TERMC)1930,1930,1940
1930 IF(TERMD)2100,1570,1570
1940 FTR1=ACOSH(XIPM/TERMC)
     IF (TERMD)1950,1960,1970
1950 FTR2=ACOS((BPM*XIPM-YPM)/SQRT(TERMA+TERMD*ZPM2))/SQRT(TERME)
     GO TO 1980
1960 FTR2=SQRT(XIPM2-TERMB)/(XIPM-YPM)
     GO TO 1980
1970 FTR2=ACOSH((BPM*XIPM-YPM)/SQRT(TERMA+TERMD*ZPM2))/SQRT(TERMD)
```

```
1980 IF (ZPM)1990,2020,1990
1990 FTR3=     (-XIPM*YPM+BPM*TERMB)/SQRT(TERMB*(TERMA+TERMD*ZPM2))
     IF (FTR3+1.0)2060,2060,2000
2000 IF (FTR3-1.0)2010,2030,2030
2010 FTR3=ACOS(FTR3)*SZ
     GO TO 2070
2020 IF (YPM)2030,2030,2040
2030 FTR3=0.
     GO TC 2070
2040 IF (XIPM-BPM*YPM)2030,2050,2060
2050 FTR3=PI/2.
     GO TO 2070
2060 FTR3=PI*SZ
2070 IF(LVSD)2080,2090,2090
2080 P=(TERMAA*FTR2+YPM*FTR1-ZPM*FTR3)/(CHORD(NCT)*PI)
     S=-CONSTD*(TERMAA*(FTR1-BPM*FTR2)+BPM*ZPM*FTR3
    1-SQRT(XIPM2-TERMB))/CHORD(NCT)
     D=-CONSTD*(ZPM*(BPM*FTR1-TERMD*FTR2)-TERMAA*FTR3)
    1/CHORD(NCT)
     GO TO 2370
2090 P=FTR2/(BETA*PI)
     S=-(FTR1-BPM*FTR2)/PI
     D=FTR3/PI
     GO TO 2370
2100 IF(YPM)1570,1570,2110
2110 IF (XIPM-TERMC)2120,2130,1940
2120 IF (XIPM-(BPM*YPM+SQRT(TERME)*ABS(ZPM)))1570,2140,2130
2130 IF(YPM-BPM*XIPM)1570,2140,2150
2140 CT=.5
     GO TO 2160
2150 CT=1.
2160 IF(LVSD)2170,2180,2180
2170 P=(TERMAA/SQRT(TERME)-ZPM*CT*SZ)/CHORD(NCT)
     S=BPM*P*BETA
     D=BETA*(TERMAA*CT*SZ-ZPM*SQRT(TERME))/CHORD(NCT)
     GO TO 2370
2180 P=CT/(SQRT(TERME)*BETA)
     S=3PM*P*BETA
     D=CT*SZ
     GO TO 2370
2190 TERMD=BPM2+1.
     TERME=SQRT(TERMD)
     FTR1=XIPM/TERMC
     IF(FTR1)2210,2200,2200
2200 FTR1=ASINH(FTR1)
     GO TO 2220
2210 FTR1=-ASINH(-FTR1)
2220 FTR2=(YPM+BPM*XIPM)/SQRT(TERMA+TERMD*ZPM2)
     IF(FTR2)2240,2230,2230
2230 FTR2=ASINH(FTR2)/TERME
     GO TO 2250
2240 FTR2=-ASINH(-FTR2)/TERME
2250 IF(ZPM)2260,2290,2260
2260 FTR3=(BPM*TERMB-XIPM*YPM)/SQRT(TERMB*(TERMA+TERMD*ZPM2))
     IF(FTR3+1.)2330,2330,2270
2270 IF(FTR3-1.)2280,2300,2300
2280 FTR3=ACOS(FTR3)*SZ
     GO TO 2340
2290 IF(YPM)2300,2300,2310
2300 FTR3=0.
     GO TO 2340
2310 IF(XIPM-BPM*YPM)2300,2320,2330
2320 FTR3=PI/2.
     GO TO 2340
2330 FTR3=PI*SZ
```

```
 2340 FTR4=SQRT(XIPM2+TERMB)
      IF(LVSD)2350,2360,2360
 2350 P=(TERMAA*FTR2+YPM*FTR1-ZPM*FTR3)/(CHORD(NCT)*2.*PI)
      S=-CONSTD*(TERMAA*(FTR1-BPM*FTR2)+BPM*ZPM*FTR3-FTR4)/
     1(2.*CHORD(NCT))
      D=-CONSTD*(ZPM*(BPM*FTR1-TERMD*FTR2)-TERMAA*FTR3)/CHORD(NCT)
      GO TO 2370
 2360 P=FTR2/(2.*BETA*PI)
      S=-(FTR1-BPM*FTR2)/(2.*PI)
      D=FTP3/PI
 2370 CONTINUE
      SIM=1.
      R(K)=R(K)+S*SINDT
      IF(THKW .NE. 1.0)  GO TO 2390
      IF(DELTAT.EQ.0.0.AND.ZPM.EQ.0.0) D=0.0
      GO TO 2400
 2390 IF(I.EQ.J.AND.ISIDE.EQ.1) P=0.
      IF(ISIDE.GT.1) GO TO 2400
      IF(IN.LE.0)GO TO 2400
      VS2(IN)=0.
      IF(J.LT.I) GO TO 2400
      NA=(IN-1)/NR+1
      NB=(JN-1)/NR+1
      IF(NA.EQ.NB) S=0.
      IF(J.EQ.I) VS1(IN)=.25*BPM1
      IF(J.GT.I) VS2(IN)=.25*(BPM1-BPM3)
 2400 C(K)=C(K)+D*COSDT*SM
      Q(K)=Q(K)-P*SM
      RPM(K)=RPM(K)+S*SM
      CPM(K)=CPM(K)+D*SM
      RPMM(K)=RPMM(K)-S*SM
      CPMM(K)=CPMM(K)-D*SM*SM
 2410 CONTINUE
 2420 CONTINUE
C
      U(J)=U(J)+Q(1)-Q(2)-Q(3)+Q(4)
      V(J)=V(J)+(R(1)-R(2)-R(3)+R(4))
      W(J)=W(J)+(C(1)-C(2)-C(3)+C(4))
      VPM(J)=VPM(J)+(RPM(1)-RPM(2)-RPM(3)+RPM(4))
      WPM(J)=WPM(J)+(CPM(1)-CPM(2)-CPM(3)+CPM(4))
      VPMM(J)=VPMM(J)+(RPMM(1)-RPMM(2)-RPMM(3)+RPMM(4))
      WPMM(J)=WPMM(J)+(CPMM(1)-CPMM(2)-CPMM(3)+CPMM(4))
 2430 CONTINUE
C
C
C          CORRECTING WING DIHEDRAL EFFECT, SEE EQ(62) OF REPORT
C
      IF(LVSD)2460,2440,2460
C
 2440 DO 2450 J=1,NPANEL
      UU(J)=U(J)
      VV(J)=VPMM(J)*COST-WPMM(J)*SINT
      WW(J)=WPM(J)*COST+VPM(J)*SINT
 2450 A(J)=V(J)+W(J)
C
      GO TO 2510
 2460 IF(IT)2490,2490,2470
C
 2470 DO 2480 J=1,NPANEL
      US(J)=U(J)
      VS(J)=VPMM(J)*COST-WPMM(J)*SINT
      WS(J)=WPM(J)*COST+VPM(J)*SINT
 2480 AS(J)=V(J)+W(J)
C
      IF(IT)2510,2510,1010
```

```
C
 2490 DO 2500 J=1,NPANEL
      UU(J)=US(J)-U(J)*SL
      VV(J)=VS(J)-(VPMM(J)*COST-WPMM(J)*SINT*ST)*SL
      WW(J)=WS(J)-(WPM(J)*COST+VPM(J)*SINT*ST)*SL
 2500 A(J)=AS(J)-(V(J)+W(J))*SL
C
      IF(LVSD)2470,2440,2510
 2510 WRITE(NTAPEA)(A(J),J=1,NPANEL)
      WRI,E(NTAPEB)(UU(J),VV(J),WW(J),J=1,NPANEL)
      IF.IT)2530,2520,2530
 2520 IF(LVSD)1120,2440,2530
C
C *****************
 2530 CONTINUE
C *****************
C
      IF(KACE-2)2580,2580,2540
 2540 IF(NCT-NPANEL)2550,2580,2580
 2550 IF(THKW)2580,2560,2580
 2560 NC=NWING/NROW(2)
      NR=NROW(2)
      I1=NBODY+1
      I2=NPANEL
      NCT=NBODY
      GO TO 1060
C
 2570 WRITE(NTAPEO,2590)
 2580 CALL FOR EOF (NTAPEA)
      CALL FOR EOF (NTAPEB)
C
C -----------------
      RETURN
C -----------------
 2590 FORMAT(1HO,41HERROR RETURN. WING LIES OUTSIDE MACH CONE)
      END
C
      SUBROUTINE INTAPE
C
C ...............................................................
C     READS IN GEOMETRICAL DATA FROM PANELING AND DEFINITION LINKS
C ...............................................................
C
      COMMON/ MAIN / NTAPEA,NTAPEB,NTAPEC,NTAPED,NTAPEE,NTAPEF,NTAPEI,
     1          NTAPEO,NBODY,NWING,XMACH,SYM,KACE,NPOLAR,IRW
      COMMON /DSXB/XBS1,XBS2,DSB1,DSB2
C
      COMMON /BLOCK/ALPHAS(210),AREA(210),A(210),ALPHAC(110),ALPHAT(110)
     1,CHORD(210)
     2,ISYM
     3,NPART(210),NPANEL,NROW(2), NWS
     4,THETA(210),TAIL,THKOMP,YAW, BROLL
     5,U(210)
     6,V(210),VPM(210),VV(210),VPMM(210)
     7,W(210),WPM(210),WW(210),WPMM(210)
     8,X(210,3,4),XBAR(210),XC(210)
     9,Y(210,3,4),YBAR(210),YC(210)
     A,Z(210,3,4),ZBAR(210),ZC(210)
      COMMON /NTHICK/THKW
      DIMENSION JSAVE(2),NSAVE(2),XB(51),R(51),ZDELTA(51)
      DIMENSION THETAB(11), THETB(11)
      DIMENSION XYZ(3)
C
C
      READ (NTAPEI,1290) XMACH, SYM, THKOMP, YAW, WING, BROLL
```

```
C
      NWS = IFIX(WING)
C
      BCOND=0
      KOND=BCOND
      IF (SYM)1010,1000,1010
 1000 ISYM=1
      GO TO 1020
 1010 ISYM=2
 1020 CONTINUE
C
C     CODE TO INDICATE IF THE CASE INCLUDES A THICK WING
      THKW = THKOMP
C
C     TEST FOR TYPE OF CONFIGURATION
      IF (NBODY)1040,1030,1040
C
C     WING ONLY CASE
 1030 KACE=1
      JSAVE(1)=0
      NSAVE(1)=NWING
      NPANEL=NWING
      NBODYS=0
      NRG=1
      GO TO 1070
C
 1040 CONTINUE
C
C     READ NUMBER OF BODY SOURCE SEGMENTS AND THE INDEX OF THE X -
C     COORDINATE AT LEADING EDGE OF THE WING
      READ (NTAPEB) (XYZ(I),I=1,3),ZA,S,T
      ALPHAA=XYZ(3)/XYZ(1)
      READ (NTAPEB) NBODYS,NXLE
C
C     READ BODY X-COORDINATES, BODY RADIUS, AND INCREMENT OF BODY CAMBER
      READ (NTAPEB) (XB(I),R(I),ZDELTA(I),I=1,NBODYS)
      REWIND NTAPEB
C
      IF (NBODY)1050,1280,1060
C
C     BODY ONLY CASE
 1050 KACE=2
      NPANEL=0
      NWING=0
      NWINGS = 0
      WRITE (NTAPEC) NBODY,NWING,XMACH,SYM,KACE,THKOMP,TAW,BROLL
C
      GO TO 1160
C
C     WING AND BODY CASE
 1060 KACE=3
      JSAVE(1)=0
      NSAVE(1)=NBODY
      JSAVE(2)=NBODY
      NSAVE(2)=NWING
      NPANEL=NWING+NBODY
      NRG=2
C
 1070 CONTINUE
C
C
      DO 1120 L=1,NRG
      J=JSAVE(L)
      N=NSAVE(L)
C
```

```
      DO 1100I=1,N
      J=J+1
C
C     READS IN PANEL COORDINATES
      READ (NTAPEC) NPN,NPART(J),(X(J,1,K),Y(J,1,K),Z(J,1,K),K=1,4)
      IF (NPART(J)-1)1080,1100,1080
 1080 NPT=NPART(J)
      DO 1090M=2,NPT
      READ (NTAPEC)(X(J,M,K),Y(J,M,K),Z(J,M,K),K=1,4)
 1090 CONTINUE
 1100 CONTINUE
C
      J=JSAVE(L)
      DO 1110I=1,N
      J=J+1
C
C     READS IN COORDINATES OF CENTROIDS, CONTROL POINTS, AREAS, THETAS,
C     ALPHAS, AND CHORD LENGTHS
      READ (NTAPEC) NPN,XBAR(J),YBAR(J),ZBAR(J),XC(J),YC(J),ZC(J),AREA(J
     1),THETA(J),ALPHAS(J),CHORD(J)
 1110 CONTINUE
C
C     READS IN THE NUMBER OF PANELS IN A COLUMN AND THE LOCATION OF THE
C     CONTROL POINT
      READ (NTAPEC) NROW(L),XCTP
 1120 CONTINUE
      IF (NBODY .NE. 0) READ (NTAPEC) NTHETB
      IF (NBODY .NE. 0) READ (NTAPEC) (THETB(N), N = 1, NTHETB)
      REWIND NTAPEC
C
      DO 1160 J=1,NWING
      JJ=J+NBODY
C
C     STORES THICKNESS SLOPES AND CAMBER SLOPES
      ALPHAT(J)=0.0
 1140 ALPHAC(J)=0.0
      NS=NBODY+1
C     WRITE 1ST FILE ON TAPEC
      WRITE (NTAPEC) NBODY,NWING,XMACH,SYM,KACE,THKOMP,YAW,BROLL
C
C
C
C     WRITES COORDINATES OF CENTROIDS, CONTROL POINTS, AREAS, THETAS,
C     ALPHAS, CHORD LENGTHS, NUMBER OF REGIONS, NUMBER OF ROWS PER
C     COLUMN IN A REGION, AND CONTROL POINT LOCATION
      WRITE (NTAPEC) (I,XBAR(I),YBAR(I),ZBAR.I),XC(I),YC(I),ZC(I),AREA(I
     1),THETA(I),ALPHAS(I),CHORD(I),I=1,NPANEL),NRG,(NROW(I),I=1,NRS),XC
     2TP
      CALL FOR EOF (NTAPEC)
C
C     WRITE THICKNESS SLOPES. 2ND FILE ON TAPEC
      WRITE (NTAPEC) (ALPHAT(I),I=1,NWING)
      CALL FOR EOF (NTAPEC)
C
C     WRITE CAMBER SLOPES. 3RD FILE ON TAPEC
      WRITE (NTAPEC) (ALPHAC(I),I=1,NWING)
      CALL FOR EOF (NTAPEC)
C
      IF (NBODY)1160,1230,1160
 1160 CONTINUE
C
C     WRITE SLOPE OF BODY AXIS WITH RESPECT TO THE DEFINING AXIS
C     FIRST PART OF 4TH FILE ON TAPEC
      WRITE (NTAPEC) ALPHAA,DSB1,DSB2,XBS1,XBS2
C
```

```
C     WRITE NUMBER OF BODY SOURCE SEGMENTS, INDEX OF X- COORDINATE AT
C     LEADING EDGE OF WING, HEIGHT OF WING PLANE ABOVE BODY AXIS, X
C     COORDINATE OF BODY.  SECOND PART OF 4TH FILE ON TAPEC.
      WRITE (NTAPEC) NBODYS,NXLE,ZA,(XB(I),I=1,NBODYS)
      IF (NWING)1180,1170,1180
 1170 CONTINUE
C
C     FOR BODY ONLY CASE, READ NUMBER OF THETAS REPRESENTING BODY
      READ (NTAPED) NDUMMY,NDUMMY,NDUMMY,NTHETB
C
C     READ BODY THETAS
      READ (NTAPED) (THETB(J),J=1,NTHETB)
      REWIND NTAPED
 1180 DO 1190 J = 1,NTHETB
 1190 THETB(J) = 57.29578*THETB(J)
      IF (NWING .EQ. 0) GO TO 1210
C     FOR WING-BODY CASE COMPUTE THETAS FROM PANEL REPRESENTATION OF BOD
      NROWB = NROW(1)
      NTHETA = NBODY/NROWB
      DO 1200 J = 1, NTHETA
      JJ = (J-1)*NROWB+1
 1200 THETAB(J) = ABS(THETA(JJ))*57.29578
C
C     WRITE NUMBER OF THETAS, AND THETAS. 3RD PART OF 4TH FILE ON C
      WRITE (NTAPEC) NTHETA,(THETAB(I),I=1,NTHETA)
 1210 WRITE (NTAPEC) NTHETB, (THETB(I),I=1,NTHETB)
C
C     WRITE RADIUS AND CAMBER INCREMENT OF BODY. LAST PART OF 4TH FILE
      WRITE (NTAPEC) (R(I),I=1,NBODYS)
      WRITE (NTAPEC) (ZDELTA(I),I=1,NBODYS)
      CALL FOR EOF (NTAPEC)
      IF (NBODY)1220,1220,1240
 1220 CONTINUE
      CALL FOR EOF (NTAPEC)
      CALL FOR EOF (NTAPEC)
 1230 CONTINUE
      CALL FOR EOF (NTAPEC)
C
C     STORE GEOMETRY DATA ON SCRATCH/SAVE TAPE
C     FOR USE IN FLOW VISUALIZATION LINK
 1240 CONTINUE
      IF (NBODY .NE.-1) NWINGS=NWING+NWING/NROW(NRG)
C     WRITE 5TH FILE ON TAPEC
      WRITE (NTAPEC) KACE,NPANEL,NBODY,NWING,NBODYS,NWINGS
     1,NROW,XMACH,STM
      CALL FOR EOF (NTAPEC)
C
      IF (NPANEL)1270,1270,1250
 1250 CONTINUE
      WRITE (NTAPEC) NPANEL
      DO 1260 J=1,NPANEL
      WRITE (NTAPEC) NPART(J)
      NPT=NPART(J)
      WRITE (NTAPEC) ((X(J,M,K),Y(J,M,K),Z(J,M,K),K=1,4),M=1,NPT)
 1260 CONTINUE
      WRITE (NTAPEC) (XBAR(J),YBAR(J),ZBAR(J),J=1,NPANEL)
      WRITE (NTAPEC) (XC(J),YC(J),ZC(J),J=1,NPANEL)
      WRITE (NTAPEC) (ALPHAS(J),THETA(J),CHORD(J),J=1,NPANEL)
 1270 CONTINUE
      CALL FOR EOF (NTAPEC)
 1280 CONTINUE
C
C
C
C -----------------
```

```
      RETURN
C ------------------
 1290 FORMAT(7F10.0)
      END
C
      OVERLAY(WOOD,4,0)
      PROGRAM REDUCE
C
C ...............................................................
C     REDUCE AERODYNAMIC INFLUENCE COEFFICIENTS MATRIX AND FORM THE
C     (D) AND (E) MATRICES
C ...............................................................
C
      COMMON/ MAIN / NTAPEA,NTAPEB,NTAPEC,NTAPED,NTAPEE,NTAPEF,NTAPEI,
     1          NTAPEO,NBODY,NWING,XMACH,SYM,KACE,NPOLAR,IRW
C
      DIMENSION AB(100),ABB(100,100),ABW(100,25),AWB(110,100),AWW(110,25
     1),E(100,25),D(110),NSIZE(5)
C
      EQUIVALENCE (ABB(1,1),AWB(1,1))
C
C
C     COMPUTE SIZE OF PARTIONS
      MAXP=25
      CALL SIZE (NWING,MAXP,NPART,NSIZE)
      DO 1000 J=1,NBODY
 1000 READ (NTAPED) (AWB(I,J),I=1,NWING)
      CALL FSF(1,NTAPED,IRR)
C
C     FORM (D) MATRIX
      DO 1020 K=1,NBODY
      READ (NTAPED) (AB(I),I=1,NBODY)
      DO 1010 I=1,NWING
      D(I)=0.0
      DO 1010 J=1,NBODY
      D(I)=D(I)+AWB(I,J)*AB(J)
 1010 CONTINUE
      WRITE (KTAPEF) (D(I),I=1,NWING)
 1020 CONTINUE
      CALL FOR EOF (NTAPEF)
      REWIND NTAPED
      CALL FSF(1,NTAPED,IRR)
      DO 1030 J=1,NBODY
 1030 READ (NTAPED) (ABB(I,J),I=1,NBODY)
C
C     COPY INVERSE BODY MATRIX
      DO 1040 J=1,NBODY
 1040 WRITE (NTAPEF) (ABB(I,J),I=1,NBODY)
C
C     MATRIX REDUCTION
      DO 1130 L=1,NPART
      NS=NSIZE(L)
      IF (L .EQ. 1) GO TO 1060
      DO 1050 J=1,NBODY
 1050 READ (NTAPED) (ABB(I,J),I=1,NBODY)
 1060 REWIND NTAPED
      DO 1070 K=1,NS
 1070 READ (NTAPEA)(ABW(I,K),I=1,NBODY),(AWW(I,K),I=1,NWING)
C
C     FORM (E) MATRIX
      DO 1080 K=1,NS
      DO 1080 M=1,NBODY
      E(M,K)=0.
      DO 1080 N=1,NBODY
      E(M,K)=E(M,K)-ABB(M,N)*ABW(N,K)
 1080 CONTINUE
      DO 1090 J=1,NBODY
 1090 READ (NTAPED)(AWB(I,J),I=1,NWING)
      CALL FSF(1,NTAPED,IRR)
      DO 1100 K=1,NS
      DO 1100 I=1,NWING
      DO 1100 M=1,NBODY
      AWW(I,K)=AWW(I,K)+AWB(I,M)*E(M,K)
 1100 CONTINUE
C
      DO 1110 K=1,NS
 1110 WRITE (NTAPEF)(E(I,K),I=1,NBODY)
      DO 1120 K=1,NS
 1120 WRITE (NTAPEE)(AWW(I,K),I=1,NWING)
 1130 CONTINUE
      REWIND NTAPEA
      REWIND NTAPED
      REWIND NTAPEE
      CALL FOR EOF (NTAPEF)
C
      REWIND NTAPEF
C
C ------------------
C ------------------
      END
C
      SUBROUTINE SIZE(MM,NMAX,NPART,NSIZE)
C
C ......................................................................
C
C ......................................................................
C
      DIMENSION NSIZE(1)
C
C     MM     SIZE OF MATRIX
C     NMAX   MAXIMUM SIZE OF PARTITIONS
C     NPART  NUMBER OF PARTITIONS
C     NSIZE  ARRAY OF THE SIZE OF EACH PARTITION
C
C
      NPMIN=2
      NT=MM
      NPART=MAX0 (NPMIN,1+(NT-1)/NMAX)
      N=NPART
 1000 NSIZE(N)=NT/N
      NT=NT-NSIZE(N)
      N=N-1
      IF (N)1010,1010,1000
C
C ------------------
 1010 RETURN
C ------------------
      END
C
      OVERLAY(WOOD,5,0)
      PROGRAM PARTY
C
C ......................................................................
C     PARTITION VELOCITY COMPONENTS MATRIX
C ......................................................................
C
      COMMON/ MAIN / NTAPEA,NTAPEB,NTAPEC,NTAPED,NTAPEE,NTAPEF,NTAPEI,
     1         NTAPEO,NBODY,NWING,XMACH,STM,KACE,NPOLAR,IRW
C
      DIMENSION U(110),V(110),W(110),UU(110),VV(110),WW(110)
```

```
C
C
      CALL FSF(1,NTAPEA,IRR)
      CALL FSF(1,NTAPEB,IRR)
      DO 1000 J=1,NBODY
      READ (NTAPEB) (U(I),V(I),W(I),I=1,NBODY),(UU(I),VV(I),WW(I),I=1,
     1NWING)
      WRITE (NTAPED) (U(I),V(I),W(I),I=1,NBODY)
      WRITE (NTAPEA) (UU(I),VV(I),WW(I),I=1,NWING)
 1000 CONTINUE
      CALL FOR EOF (NTAPED)
      REWIND NTAPEA
      CALL FSF(1,NTAPEA,IRR)
C
      DO 1010 J=1,NBODY
      READ (NTAPEA) (UU(I),VV(I),WW(I),I=1,NWING)
      WRITE (NTAPED) (UU(I),VV(I),WW(I),I=1,NWING)
 1010 CONTINUE
C
      CALL FOR EOF (NTAPED)
      REWIND NTAPEA
      CALL FSF(1,NTAPEA,IRR)
C
      DO 1020 J=1,NWING
      READ (NTAPEB) (U(I),V(I),W(I),I=1,NBODY),(UU(I),VV(I),WW(I),I=1,NW
     1ING)
      WRITE (NTAPED) (U(I),V(I),W(I),I=1,NBODY)
      WRITE (NTAPEA) (UU(I),VV(I),WW(I),I=1,NWING)
 1020 CONTINUE
C
      REWIND NTAPEA
      REWIND NTAPEB
      CALL FOR EOF (NTAPED)
      CALL FSF(1,NTAPEA,IRR)
C
      DO 1030 J=1,NWING
      READ (NTAPEA) (UU(I),VV(I),WW(I),I=1,NWING)
      WRITE (NTAPED) (UU(I),VV(I),WW(I),I=1,NWING)
 1030 CONTINUE
      CALL FOR EOF (NTAPED)
      REWIND NTAPEA
      REWIND NTAPED
C
C ------------------
C ------------------
      END
C
      OVERLAY(WOOD,6,0)
      PROGRAM WOOD60
      COMMON/ MAIN / NTAPEA,NTAPEB,NTAPEC,NTAPED,NTAPEE,NTAPEF,NTAPEI,
     1          NTAPEO,NBODY,NWING,XMACH,SYM,KACE,NPOLAR,IRW
      COMMON /TRNSFR/ IOVR10,IOVR60,IOVR70
      COMMON /COM60/ ISAVET
      IF (IOVR60.EQ.2) GO TO 100
      CALL SAVTAP(ISAVET)
      GO TO 999
  100 CALL USETAP
 999  CONTINUE
      END
C
      SUBROUTINE SAVTAP(ISAVET)
C
C ..................................................................
C     WRITES AERODYNAMIC MATRICES ON A LOGICAL TAPE FOR FUTURE USE
C ..................................................................
```

```
C
      COMMON/ MAIN / NTAPEA,NTAPEB,NTAPEC,NTAPED,NTAPEE,NTAPEF,NTAPEI,
     1              NTAPEO,NBODY,NWING,XMACH,SYM,KACE,NPOLAR,IRW
      COMMON /NTHICK/THKW
      DIMENSION A(210),B(210),C(210)
      DIMENSION DUM(10),NROW(2)
C
      IF (ISAVET)1010,1010,1000
 1000 READ (NTAPEC)NBODY,NWING,XMACH,SYM,KACE,THKW,YAW,BROLL
C
C ****************
      GO TO (1020,1010,1020),KACE
C
 1010 CALL FSF(6,NTAPEC,IRR)
      GO TO 1090
 1020 NPANEL=NBODY+NWING
      READ (NTAPEC) ( M, ( DUM(J), J=1,10), I=1,NPANEL), NRG,
     1(NROW(I),I=1,NRG),DUMMY
      NCOLW=NWING/NROW(NRG)
      CALL FSF( 6,NTAPEC,IRR)
      GO TO (1030,1090,1060),KACE
 1030 IF (THKW .EQ. 0.) GO TO 1040
      CALL TTAPE (0,NWING,NWING+NCOLW,NTAPEA,NTAPEC,A,B,C)
 1040 CALL FSF(1,NTAPEA,IRR)
      CALL TTAPE(0,NWING,NWING,NTAPEA,NTAPEC,A,B,C)
      CALL FSF(1,NTAPEA,IRR)
      CALL TTAPE(0,NWING+1,NWING+1,NTAPEA,NTAPEC,A,B,C)
      CALL FSF(1,NTAPEA,IRR)
      CALL TTAPE(0,NWING+2,NWING+2,NTAPEA,NTAPEC,A,B,C)
      REWIND NTAPEA
      IF (THKW .EQ. 0.) GO TO 1050
      CALL TTAPE(1,NWING,NWING+NCOLW,NTAPEB,NTAPEC,A,B,C)
 1050 CALL FSF(1,NTAPEB,IRR)
      CALL TTAPE(1,NWING,NWING,NTAPEB,NTAPEC,A,B,C)
      REWIND NTAPEB
      CALL TTAPE(0,NWING,NWING,NTAPEE,NTAPEC,A,B,C)
      CALL FSF(1,NTAPEE,IRR)
      CALL TTAPE(0,NWING,NWING,NTAPEE,NTAPEC,A,B,C)
      REWIND NTAPEE
      GO TO 1090
C
 1060 IF (THKW .EQ. 0.) GO TO 1070
      CALL TTAPE(0,NPANEL,NWING +NCOLW,NTAPEA,NTAPEC,A,B,C)
 1070 CALL FSF(1,NTAPEA,IRR)
      CALL TTAPE(0,NWING,NWING,NTAPEA,NTAPEC,A,B,C)
      CALL FSF(1,NTAPEA,IRR)
      CALL TTAPE(0,NWING+1,NWING+1,NTAPEA,NTAPEC,A,B,C)
      CALL FSF(1,NTAPEA,IRR)
      CALL TTAPE(0,NWING+2,NWING+2,NTAPEA,NTAPEC,A,B,C)
      REWIND NTAPEA
      IF (THKW .EQ. 0.) GO TO 1080
      CALL TTAPE(1,NPANEL,NWING +NCOLW,NTAPEB,NTAPEC,A,B,C)
 1080 REWIND NTAPEB
      CALL TTAPE(1,NBODY,NBODY,NTAPED,NTAPEC,A,B,C)
      CALL FSF(1,NTAPED,IRR)
      CALL TTAPE(1,NWING,NBODY,NTAPED,NTAPEC,A,B,C)
      CALL FSF(1,NTAPED,IRR)
      CALL TTAPE(1,NBODY,NWING,NTAPED,NTAPEC,A,B,C)
      CALL FSF(1,NTAPED,IRR)
      CALL TTAPE(1,NWING,NWING,NTAPED,NTAPEC,A,B,C)
      REWIND NTAPED
      CALL TTAPE(0,NWING,NWING,NTAPEE,NTAPEC,A,B,C)
      CALL FSF(1,NTAPEE,IRR)
      CALL TTAPE(0,NWING,NWING,NTAPEE,NTAPEC,A,B,C)
      REWIND NTAPEE
      CALL TTAPE(0,NWING,NBODY,NTAPEF,NTAPEC,A,B,C)
      CALL FSF(1,NTAPEF,IRR)
      CALL TTAPE(0,NBODY,NBODY,NTAPEF,NTAPEC,A,B,C)
      CALL TTAPE(0,NBODY,NWING,NTAPEF,NTAPEC,A,B,C)
      REWIND NTAPEF
C
 1090 CONTINUE
      CALL FOR EOF (NTAPEC)
      REWIND NTAPEC
C
C ------------------
      RETURN
C ------------------
      END
C
      SUBROUTINE USETAP
C
C ...................................................................
C       READS AERODYNAMIC MATRICES FROM SAVE TAPE AND REWRITES
C ...................................................................
C
      COMMON/ MAIN / NTAPEA,NTAPEB,NTAPEC,NTAPED,NTAPEE,NTAPEF,NTAPEI,
     1          NTAPEO,NBODY,NWING,XMACH,SYM,KACE,NPOLAR,IRW
      COMMON /WTHICK/THKW
      DIMENSION A(210),B(210),C(210)
      DIMENSION DUM(10),NROW(2)
C
C
      READ(NTAPEC)NBODY,NWING,XMACH,SYM,KACE,THKW,YAW,BROLL
C
      GO TO (1000,1070,1000),KACE
 1000 NPANEL=NBODY+NWING
      READ (NTAPEC) ( M, ( DUM(J), J=1,10), I=1,NPANEL), NRG,
     1(NROW(I),I=1,NRG),DUMMY
      NCOLW=NWING/NROW(NRG)
      CALL FSF( 6,NTAPEC,IRR)
C
      GO TO (1010,1070,1040),KACE
 1010 IF (THKW .EQ. 0.) GO TO 1020
      CALL TTAPE(0,NWING,NWING+NCOLW,NTAPEC,NTAPEA,A,B,C)
 1020 CALL FOR EOF (NTAPEA)
      CALL TTAPE(0,NWING,NWING,NTAPEC,NTAPEA,A,B,C)
      CALL FOR EOF (NTAPEA)
      CALL TTAPE(0,NWING+1,NWING+1,NTAPEC,NTAPEA,A,B,C)
      CALL FOR EOF (NTAPEA)
      CALL TTAPE(0,NWING+2,NWING+2,NTAPEC,NTAPEA,A,B,C)
      CALL FOR EOF (NTAPEA)
      REWIND NTAPEA
      IF (THKW .EQ. 0.) GO TO 1030
      CALL TTAPE(1,NWING,NWING+NCOLW,NTAPEC,NTAPEB,A,B,C)
 1030 CALL FOR EOF (NTAPEB)
      CALL TTAPE(1,NWING,NWING,NTAPEC,NTAPEB,A,B,C)
      CALL FOR EOF (NTAPEB)
      REWIND NTAPEB
      CALL TTAPE(0,NWING,NWING,NTAPEC,NTAPEE,A,B,C)
      CALL FOR EOF (NTAPEE)
      CALL TTAPE(0,NWING,NWING,NTAPEC,NTAPEE,A,B,C)
      CALL FOR EOF (NTAPEE)
      REWIND NTAPEE
      GO TO 1070
 1040 IF (THKW .EQ. 0.) GO TO 1050
      CALL TTAPE(0,NPANEL,NWING +NCOLW,NTAPEC,NTAPEA,A,B,C)
 1050 CALL FOR EOF (NTAPEA)
      CALL TTAPE(0,NWING,NWING,NTAPEC,NTAPEA,A,B,C)
      CALL FOR EOF (NTAPEA)
      CALL TTAPE(0,NWING+1,NWING+1,NTAPEC,NTAPEA,A,B,C)
      CALL FOR EOF (NTAPEA)
      CALL TTAPE(0,NWING+2,NWING+2,NTAPEC,NTAPEA,A,B,C)
      CALL FOR EOF (NTAPEA)
      REWIND NTAPEA
      IF (THKW .EQ. 0.) GO TO 1060
      CALL TTAPE(1,NPANEL,NWING +NCOLW,NTAPEC,NTAPEB,A,B,C)
 1060 CALL FOR EOF (NTAPEB)
      REWIND NTAPEB
      CALL TTAPE(1,NBODY,NBODY,NTAPEC,NTAPED,A,B,C)
      CALL FOR EOF (NTAPED)
      CALL TTAPE(1,NWING,NBODY,NTAPEC,NTAPED,A,B,C)
      CALL FOR EOF (NTAPED)
      CALL TTAPE(1,NBODY,NWING,NTAPEC,NTAPED,A,B,C)
      CALL FOR EOF (NTAPED)
      CALL TTAPE(1,NWING,NWING,NTAPEC,NTAPED,A,B,C)
      CALL FOR EOF (NTAPED)
      REWIND NTAPED
      CALL TTAPE(0,NWING,NWING,NTAPEC,NTAPEE,A,B,C)
      CALL FOR EOF (NTAPEE)
      CALL TTAPE(0,NWING,NWING,NTAPEC,NTAPEE,A,B,C)
      CALL FOR EOF (NTAPEE)
      REWIND NTAPEE
      CALL TTAPE(0,NWING,NBODY,NTAPEC,NTAPEF,A,B,C)
      CALL FOR EOF (NTAPEF)
      CALL TTAPE(0,NBODY,NBODY,NTAPEC,NTAPEF,A,B,C)
      CALL TTAPE(0,NBODY,NWING,NTAPEC,NTAPEF,A,B,C)
      CALL FOR EOF (NTAPEF)
      REWIND NTAPEF
 1070 CONTINUE
      REWIND NTAPEC
C
C ------------------
      RETURN
C ------------------
      END
C
      SUBROUTINE TTAPE(KIND,IN,JN,NTIN,NTOUT,A,B,C)
C
C ....................................................................
C     COPIES MATRICES FROM TAPE TO TAPE
C ....................................................................
C
      DIMENSION A(1),B(1),C(1)
C
      IF(KIND)1020,1000,1020
 1000 CONTINUE
      DO 1010 J=1,JN
      READ(NTIN)(A(I),I=1,IN)
 1010 WRITE(NTOUT)(A(I),I=1,IN)
C
      GO TO 1040
 1020 CONTINUE
      DO 1030 J=1,JN
      READ(NTIN)(A(I),B(I),C(I),I=1,IN)
 1030 WRITE(NTOUT)(A(I),B(I),C(I),I=1,IN)
C
 1040 CONTINUE
C
C ------------------
      RETURN
C ------------------
      END
C
      OVERLAY(WANG, 7, 0)
      PROGRAM WANG70
C
C
      COMMON/ MAIN / NTAPEA,NTAPEB,NTAPEC,NTAPED,NTAPEE,NTAPEF,NTAPEI,
     1           NTAPEO,NBODY,NWING,XMACH,SYM,KACE,NPOLAR,IRW,IPLOT
      COMMON / BODYSP/ DRDX(51), DZDX(51), DRDXC(51), XWROOT, XWTIP
      COMMON /AVAR/  A(210),ACB(21),ABX(100),AWX(110),AREA(210),ARN(2),A
     1               RWT(20),ALPHA(210),ALPHAB(210),ALPHAT(110),
     2               ALPHAW(110),ALPHAX(110),AWS(110),ABT(100),AN1N(
     3               210,2),ALPHAA,ALPHAD,ARA,ARADEG,ARB,ARW(10), ARAS,
     4               AT, AAA
      COMMON /BODY/ NUMBOD,XNOS(3),YNOS(3),ZNOS(3),NBSTAT(3)
      COMMON /BVAR/  B(210), BBCL, BBCD, BBCY, BBCM, BBCMR, BBCMY
      COMMON /CVAR/  C(210),CHORD(210),CL(210),CPNN(1,1,1),CLS(210)
     1               ,CDR,CASE,CPCALC,CBAR,CONSNT,CLBAR,CLX,CLM,CDN
      COMMON /DVAR/  D(210),DZDXB(51), DADEG, DARAD
      COMMON /IVAR/  IPOLAR
      COMMON /KVAR/  KASE,KONFIG,KPOLAR
      COMMON /NVAR/  NFMT(9),NROW(2)
     1               ,NS,KPANEL,NACEL,NROWB,NROWW,NCOLW,NTHETA,NTHETS,
     2                    NXLE,NRG,NPOLR ,NCLX
      COMMON /PVAR/  POLAR
      COMMON /RVAR/  R(51),RN(51,2), RATIOX, RFAREA
      COMMON /SVAR/  SEMIS,SLC
      COMMON /TVAR/  TITLE(20),THETA(210),THETAB(15),THETAS(15),THETAN(
     1            11,2),TCL(11),TCD(11),TCM(11),THICK,TWIST
      COMMON /UVAR/  UBWT(150),UWBT(110),UWWT(110),UNCL(1,1)
      COMMON /VVAR/  VBWT(150),VWBT(110),VWWT(110),VNCL(1,1)
     1               ,VOUT
      COMMON /WVAR/  WBWT(150),WWBT(110),WWWT(110),WNCL(1,1)
      COMMON /XVAR/  XBAR(210),XC(210),YNI(2),XNXN(2),XNTN(2),XB(51),XN(
     1               51,2),XNNCD(2),XNNCL(2),XNNCM(2),XYZ(3),XCL(11),XCD
     2               (11), XNACEL, XP, XCPBAR
      COMMON /YVAR/  YBAR(210),YC(210),YNI(2)
      COMMON /ZVAR/  ZBAR(210),ZC(21C),ZNI(2),ZDELTA(51),ZDN(51,2)
     1               ,ZP,ZA
      COMMON/PBDATA/ NBODYS,XMIDB(51), NTHETB,THETB(15),CPBB(51,15)
     1          ,XCB(3,700),NG(350),NELM(700),NE,NGRIDP,PINF,IPUNCH, CPB
     2          , SID(20)
      COMMON'ELMCNT/ XCI(3, 350)
      COMMON/WINGOT/ CPU, CPL,NWG,ISOLID,IFORM(10),YAW,BROLL,SLOPE(51)
      COMMON/EXTRA/ ARAN, DSB1, DSB2, DS1, DS2, NB1,NB2,NS1,NS2,
     1          NTN, NXN, XNOSE
      COMMON/TRNSFR/ IOVR10, IOVR60, IOVR70
      COMMON/CORTRN/ NSYM,NTRANS(20),ISYM(20),VA(3,2,20), BTFM(9,2,20)
      COMMON/NSLOPE/ DRNDX(51,2), DZNDX(51,2)
      COMMON/MATRIX/ TAX(115,115)
      COMMON/PANVEL/ VS1(110), VS2(110), WSS(110)
      COMMON/REPEAT/ IREPT
C
      DIMENSION XNS1(2), XNS2(2), NS1(2), NS2(2), DS1(2), DS2(2)
      DIMENSION CPB(150), CPU(110), CPL(110)
      INTEGER TITLE
C
      IF ( XMACH .LT. 2.5 ) GO TO 50
C
C        HIGH SUPERSONIC CASE, MACH .GT. 2.0
C
      CALL OVERLAY(4HWANG, 7, 13)
C
C        PRESSURE INTERPOLATION FOR THE BODY
C
      NSYM = 1
      CALL OVERLAY(4HWANG, 7, 8)
C
```

```
C        PRESSURE INTERPOLATION FOR THE WING
C
      NSYM = 2
      CALL OVERLAY(4HWANG, 7, 10)
      GO TO 600
C
   50 CONTINUE
C
C        RESTART RUN IRW = 2
C
      IF ( IRW  .EQ. 2 ) GO TO 100
C
      IGOTO = IOVR70 + 1
      GO TO ( 100, 200, 300, 400, 500 ) , IGOTO
C
  100 CONTINUE
      CALL FORCES
      GO TO 600
C
  200 CONTINUE
      CALL OVERLAY(4HWANG, 7, 1)
      GO TO 600
C
  300 CONTINUE
      CALL OVERLAY(4HWANG, 7, 2)
      GO TO 600
C
  400 CONTINUE
      CALL OVERLAY(4HWANG, 7, 3)
      GO TO 600
C
  500 CONTINUE
      CALL OVERLAY(4HWANG, 7, 4)
C
  600 CONTINUE
C
C
      END
C
      SUBROUTINE FORCES
C
C ....................................................................
C     C'NTROL ROUTINE FOR FORCES LINK
C ....................................................................
C
      COMMON/ MAIN / NTAPEA,NTAPEB,NTAPEC,NTAPED,NTAPEE,NTAPEF,NTAPEI,
     1         NTAPEO,NBODY,NWING,XMACH,SYM,KACE,NPOLAR,IRW,IPLOT
      COMMON / BODYSP/ DRDX(51), DZDX(51), DRDXC(51)
      COMMON /AVAR/  A(210),ACB(21),ABX(100),AWX(110),AREA(210),ARN(2),A
     1               RWT(20),ALPHA(210),ALPHAB(210),ALPHAT(110),
     2               ALPHAW(110),ALPHAX(110),AWS(110),ABT(100),ANIN(
     3               210,2),ALPHAA,ALPHAD,ARA,ARADEG,ARB,ARW(10), ARAS,
     4               AT, AAA
      COMMON /BODS/ NUMBOD,XNOS(3),YNOS(3),ZNOS(3),NBSTAT(3)
      COMMON /BVAR/  B(210), BBCL, BBCD, BBCT, BBCM, BBCMR, BBCMY
      COMMON/CVAR/C(210), CHORD(210),CL(210),CPNN(1,1,1)
     1               ,CLS(210)
     2               ,CDR,CASE,CPCALC,CBAR,CONSNT,CLBAR,CLX,CLM,CDM
      COMMON/DVAR/ D(210),DZDXB(51), DADEG,DARAD
      COMMON /IVAR/  IPOLAR
      COMMON /KVAR/  KASE,KONFIG,XPOLAR
      COMMON /NVAR/  NFMT(9),NROW(2)
     1         ,NS,NPANEL,NACEL,NROWB,NROWW,NCOLW,NTHETA,NTHETS,
     2         NXLE, NRG, NPOLR,  NCLX
      COMMON /PVAR/  POLAR
      COMMON/RVAR/ R(51), RN(51,2), RATIOX, RFAREA
      COMMON /SVAR/  SEMIS,SLC
      COMMON/TVAR/ TITLE(20), THETA(210), THETAB(15),THETAS(15), THETAN
     1        (11,2), TCL(11), TCD(11), TCM(11), THICK,TWIST
      COMMON /UVAR/  UBWT(150),UWBT(110),UWWT(110),UNCL(1,1)
      COMMON /VVAR/  VBWT(150),VWBT(110),VWWT(110),VNCL(1,1)
     1               ,VOUT
      COMMON /WVAR/  WBWT(150),WWBT(110),WWWT(110),WNCL(1,1)
      COMMON/XVAR/ XBAR(210), XC(210), XNI(2), XNXN(2), XNTN(2), XB(51),
     1        XN(51,2), XNNCD(2), XNNCL(2), XNNCM(2), XYZ(3), XCL(11),
     2        XCD(11), XNACEL, XP, XCPBAR
      COMMON /YVAR/  YBAR(210),YC(210),YNI(2)
      COMMON/ZVAR/ ZBAR(210), ZC(210), ZNI(2), ZDELTA(51), ZDN(51,2)
     1               ,ZP,ZA
      COMMON/PBDATA/ NBODYS, XMIDB(51), NTHETB, THETB(15), CPBB(51,15)
     1        ,XCB(3,700),NG(350),NELM(700),NE,NGRIDP,PINF,IPUNCH
     2       ,CPB, SID(20)
      COMMON/WINGOT/ CPU, CPL,NWG,ISOLID,IFORM(10),YAW,BROLL
      COMMON/EXTRA/ ARAN, DSB1, DSB2, DS1, DS2, NB1, NB2, NS1, NS2,
     1        NTN, NXN, XNOSE
      COMMON/CORTRN/ NSYM,NTRANS(20),ISYM(20),VA(3,2,20), BTFR(9,2,20)
      COMMON/NSLOPE/ DRNDX(51,2), DZNDX(51,2)
      COMMON/REPEAT/ IREPT
C
      DIMENSION XNS1(2),XNS2(2),NS1(2),NS2(2),DS1(2),DS2(2)
     1         , CPB(150), CPU(110), CPL(110)
      INTEGER TITLE
C
C
C     DEFINE CONSTANTS AND READ INPUT DATA
C     KONFIG = 1      WING ALONE CONFIGURATION
C     KONFIG = 2      BODY ALONE CONFIGURATION
C     KONFIG = 3      WING-BODY CONFIGURATION
C
C     CASE = 1.       CALCULATE ALPHA, GIVEN CL
C     CASE = 2.       CALCULATE CL, GIVEN ALPHA
C     CASE = 3.       OPTIMIZE WING
C
C     CPCALC =0.      LINEAR CP
C     CPCALC =1.      NON-LINEAR CP
C     CPCALC =2.      EXACT CP
C
C     POLAR = 0.      POLARS NOT REQUESTED,OMIT DELTA ALPHA CARDS
C     POLAR = 1.      POLARS REQUESTED,INSERT DELTA ALPHA CARDS
C
C     THICK =0.       WING THICKNESS PRESSURES NOT TO BE ADDED
C     THICK =1.       WING THICKNESS PRESSURES TO BE ADDED
C
C     VOUT = 0.       VELOCITY COMPONENTS NOT TO BE PRINTED
C     VOUT = 1.       VELOCITY COMPONENTS TO BE PRINTED
C
C     XNACEL =0.      EFFECTS OF NACELS NOT INCLUDED
C     XNACEL =X.      EFFECTS OF X PAIRS OF NACELLES, GEOMETRY EXPECTED
C     XNACEL =-X.     EFFECTS OF X PAIRS OF NACELLES, GEOMETRY OF SECOND
C                     PAIR SAME AS FIRST PAIR
C
C     RFAREA =0.      WING REFERENCE AREA TO BE CALCULATED
C     RFAREA =X.      WING REFERENLE AREA TO BE SPECIFIED
C
C     XP,ZP           X AND Z COORDINATES OF POINT ABOUT WHICH MOMENTS
C                     ARE TO BE COMPUTED
C
C     CBAR            REFERENCE CHORD LENGTH USED IN PITCHING
C                     MOMENT CALCULATIONS
C                     CBAR =1. IF NOT SPECIFIED
```

```
C
C     SEMIS           WING SEMI-SPAN USEDIN SPANWISE CD AND CL
C                     'ALCULATIONS
C                     :IMIS =1. IF NOT SPECIFIED
C
C
      KONFIG=KACE
      NPANEL=NWING+NBODY
      NS=NBODY+1
C
      CDR=57.2957795
      IF ( IRW  .EQ. 2 ) GO TO 1111
      IF (KONFIG-2)1000,1010,1000
C
C     READ CONFIGURATION GEOMETRY
 1000 CONTINUE
      READ  (NTAPEC) NBODY,NWING,XMACH,SYM,KACE,THK,YAW,BROLL
C
      READ (NTAPEC) (J,XBAR(I),YBAR(I),ZBAR(I),XC(I),YC(I),ZC(I),AREA(I
     1),THETA(I),DUMMY,CHORD(I),I=1,NPANEL),NRG,(NROW(I),I=1,NRG),RATIOX
C
      REWIND NTAPEC
C
 1010 CONTINUE
C
      NACEL=0
C
C     READ CARD 5A
      READ (NTAPEI,1700)CASE, CPCALC, POLAR, THICK, CAMB, XNACEL, PINF,
     1    PUNCH,SOLID,WING, ( SID(K), K=1,20), (IFORM(K), K=1,10),XP
      IPUNCH = IFIX( PUNCH )
      ISOLID = IFIX( SOLID )
      ICAMBR = IFIX( CAMB )
      NWG = IFIX( WING )
      ANCEL=ABS(XNACEL)
      NACEL=ANCEL
      NUMBOD=0
      IF(NBODY.NE.0) NUMBOD=1
      IPOLAR=0
      POLAR = POLAR + 1
      IF (CASE)1020,1680,1020
 1020 CONTINUE
C
      IF (CBAR .EQ. 0.) CBAR=1.
      IF (SEMIS .EQ. 0.)SEMIS=1.
C
      IF (VOUT)1130,1150,1130
 1130 CONTINUE
      DO 1140 J=1,9
 1140 NFMT(J)=J
      GO TO 1170
 1150 CONTINUE
      DO 1160 J=1,9
 1160 NFMT(J)=0
 1170 CONTINUE
      KASE=CASE
C
 1180 IF (KONFIG-2)1200,1190,1200
 1190 RFAREA=1.0
      GO TO 1220
 1200 RFAREA=0.0
      DO 1210 J=NS,NPANEL
 1210 RFAREA=RFAREA+AREA(J)
 1220 CONTINUE
C
```

```
C     WRITE CASE DATA
      CALL INOUT(NTAPEO,KASE,CPCALC,POLAR,THICK,VOUT,XMACH,SEMIS,
     1XP,ZP,CBAR,RFAREA,SYM)
C
C
C     ANY CONFIGURATION
      IF (KONFIG .EQ. 1) GO TO 1280
C
C     BODY ALONE OR WING-BODY CONFIGURATION
      NROWB=NROW(1)
C
      IF (KONFIG-2)1240,1240,1250
C
C     BODY ALONE CONFIGURATION
 1240 READ (NTAPEC) NBODY,NWING,XMACH,SYM,KACE
      GO TO 1260
C
C     WING-BODY CONFIGURATION
 1250 CALL FSF(3,NTAPEC,IRR)
 1260 CONTINUE
      READ (NTAPEC) ALPHAA,DSB1,DSB2,XBS1,XBS2
      NB1 = XBS1
      NB2 = XBS2
      READ (NTAPEC) NBODYS,NXLE,ZA,(XB(I),I=1,NBODYS)
      IF (KONFIG .EQ. 2) GO TO 1270
      READ (NTAPEC) NTHETA,(THETAB(I),I=1,NTHETA)
C
C     INPUT BODY VARIABLES
 1270 READ (NTAPEC) NTHETB,(THETB(I),I=1,NTHETB)
C
C     READ CARD 7A
      READ ( NTAPEI,1700) ARB, DADEG
C     READ CARD 8A
      READ ( NTAPEC ) ( R(I), I=1,NBODYS )
C     READ CARD 9A
      READ ( NTAPEC ) ( ZDELTA(I), I=1,NBODYS )
      REWIND NTAPEC
C
      RBMAX = R(1)
      DO 1271 I=2,NBODYS
      RBMAX = AMAX1( RBMAX, R(I) )
 1271 CONTINUE
C
      IF ( KONFIG  .EQ. 2 ) RFAREA = 3.14159 * RBMAX**2
      IF ( KONFIG  .EQ. 2 ) WRITE (NTAPEO,1780)  RFAREA
C
      ARA=-ALPHAA*ARB/CDR
      WRITE (NTAPEO,1770) ZA
C
      IF (KONFIG .EQ. 2) GO TO 1290
C
C     WING-BODY CONFIGURATION
      CALL BCAM(NBODYS,NROWB,XB,ZDELTA,XC,DZDXB,ACB)
C
C
C     WING ALONE OR WING-BODY CONFIGURATION
 1280 CONTINUE
      NROWW=NROW(NRG)
      NCOLW=NWING/NROWW
C
      IF (KONFIG .EQ. 3) GO TO 1290
C
C     WING ALONE CONFIGURATION
      ALPHAA=0.
      ARA=0.
```

```
C
C      ANY CONFIGURATION
 1290 CONTINUE
      CALL FSF(7,NTAPEC,IRR)
      WRITE (NTAPEC) THICK,ARA
      CALL FOR EOF (NTAPEC)
      REWIND NTAPEC
      ALPHAD=ALPHAA*CDR
      ARADEG=ARA*CDR
      YAWDEG = YAW * CDR
      ROLDEG = BROLL * CDR
      WRITE (NTAPEO,1750) ALPHAD
      WRITE (NTAPEO,1760) ARADEG
      WRITE (6,1765) YAWDEG
      WRITE (6,1766) ROLDEG
C
      IF (KONFIG.EQ.2) GO TO 1400
C
C      WING ALONE OR WING-BODY CONFIGURATION
C      INPUT WING VARIABLES
      GO TO (1300,1320,1370),KASE
C
C      READ CARD 10A
 1300 CONTINUE
      ICAM =2
      CALL READ(NTAPEI,NTAPEO,NTAPEC,NWING,CL(NS),KACE,ICAM,NROW)
      DO 1310 J=NS,NPANEL
 1310 CL(J)=-CL(J)
      GO TO 1390
C
C      READ CARD 11A
 1320 CONTINUE
      IF ( KONFIG  .EQ. 1 ) READ (NTAPEI, 1700) DUM, DADEG
      READ (NTAPEI,1700) (ARW(I), I=1,10), TWIST, NSWNG
      IF (TWIST .NE. 0.) GO TO 1340
      DO 1330 J=1,NCOLW
 1330 ARWT(J)=0.0
      GO TO 1350
C
C      READ CARD 11AA
 1340 READ (NTAPEI,1700) (ARWT(I),I=1,NCOLW)
 1350 CONTINUE
      CALL FSF (2,NTAPEC,IRR)
C
C
C
      ICAMBB = 2
      CALL READ(NTAPEI,NTAPEO,NTAPEC,NW.NG,ALPHA,KACE,ICAMBB,NROW)
      REWIND NTAPEC
C
C        READ IN WING CAMBER ( ALPHAC ) AS ALPHA
C
      IF (ICAMBR .EQ. 1 ) READ (NTAPEI,1700) (ALPHA(I), I=1,NWING)
C
      NAR = NWING / NWG
      NWPW = NWING / NWG
C
C             INCORPORATE THE EFFECT OF DIHEDRAL (THETA) AND ROLL (BROLL)
C            NWPW  NO. OF PANELS PER WING
C
      COSA = COS(ARA)
      SINA = SIN(ARA)
      TANA = TAN(ARA)
      TANY = TAN(YAW)
      A1 = 1.0/SQRT(1.0 + TANA**2 + TANY**2 )
      COSB = A1*SQRT( 1.0 + TANA**2 )
      SINB = SQRT( 1.0 - COSB**2 )
      DO 1360 J=1,NWING
      JJ=(J-1)/NROWW+1
      JW=J+NBODY
      JC = ( J - 1 )/ NAR + 1
      NWW = (J-1) / NWPW + 1
      COST = COS( THETA(JW) + BROLL )
      SINT = SIN( THETA(JW) + BROLL )
      ANGATT = ASIN( SINT*SINB*COSA + COST*SINA )
 1360 ALPHAW(J)=ALPHA(J)-ANGATT-(ARW(JC)+ARWT(JJ))/CDR
      GO TO 1390
C
C     READ CARD 13A
 1370 READ (NTAPEI,1700) CONSNT,CLBAR,XCPBAR
      IF (CONSNT .NE. 0.) GO TO 1380
      WRITE (NTAPEO,1730) CLBAR
      GO TO 1390
 1380 WRITE (NTAPEO,1730) CLBAR
      WRITE (NTAPEO,1740) XCPBAR
 1390 CONTINUE
C
C     WING ALONE OR WING-BODY CONFIGURATION
      IF (THICK)1430,1400,1430
C
 1400 CONTINUE
      CALL FSF(8,NTAPEC,IRR)
      CALL FOR EOF (NTAPEC)
      REWIND NTAPEC
      IF (KONFIG.EQ.2) GO TO 1480
C
      DO 1420 J=1,NPANEL
 1420 ALPHA(J)=0.
      DO 1425 J=1,NWING
 1425 AWS(J)=0.
      IF (KONFIG.EQ.1) GO TO 1460
      GO TO 1440
C
C     WING ALONE OR WING-BODY CONFIGURATION
 1430 CONTINUE
C
C     READ CARD 14A
      CALL TVEL(A,B,C,D,ALPHAT,UBWT,VBWT,WBWT,UWWT,VWWT,WWWT,
     1ALPHA,CHORD,THICK,NROW)
      REWIND NTAPEA
      REWIND NTAPEB
C
      DO 1455 J=1,NWING
      JJ=J+NBODY
 1455 AWS(J)=ALPHA(JJ)
      IF (KONFIG .EQ. 1) GO TO 1460
C
C     WING-BODY CONFIGURATION
C
 1278 CONTINUE
      PI = 3.14159265359
      COSA = COS(ARA)
      SINA = SIN(ARA)
      COSY = COS(YAW)
      SINY = SIN(YAW)
      NBD = NBODY
 1440 DO 1450 I = 1, NTHETA
      DO 1450 K = 1, NROWB
      J = K + (I-1)*NROWB
      SINT = SIN( THETA(J) )
      COST = COS( THETA(J) )
      SIND = SIN( DRDXC(K) )
      COSD = COS( DRDXC(K) )
      ALPHAB(J) = ACOS( -COSA*COSY*SIND + COSA*SINY*SINT*COSD
     1             + SINA*COST*COSD ) - 0.5*PI
      ABX(J)=-ALPHA(J)
      ABT(J)=ABX(J)
      IF ( YAW .LT. 0.01  .AND. BROLL .LT. 0.01 ) GO TO 1450
      J2 = NBODY + K - I*NROWB
      SINT = SIN( THETA(J) - PI )
      COST = COS( THETA(J) - PI )
      SIND = SIN( DRDXC(K) )
      COSD = COS( DRDXC(K) )
      NBD = 2*NBODY
      ALPHAB(J+NBODY) = ACOS( -COSA*COST*SIND + COSA*SINT*SINT*COSD
     1             + SINA*COST*COSD ) - 0.5*PI
 1450 CONTINUE
C
C     ANY CONFIGURATION
 1460 CONTINUE
      DO 1470 J=1,NWING
      UWBT(J)=0.
      VWBT(J)=0.
      WWBT(J)=0.
 1470 ALPHAX(J)=0.
C
 1480 CONTINUE
      CALL FSF(9,NTAPEC,IRR)
      WRITE(NTAPEC) DUMMY
      IF ( KONFIG  .EQ. 1 ) GO TO 1620
C
C     BODY ALONE OR WING-BODY CONFIGURATION
C
 1490 XNOSE = XB(1)
      IF(XMACH-1.)1500,1620,1510
 1500 CALL OVERLAY(4HWANG, 7, 5)
      GO TO 1560
C
 1510 CALL OVERLAY(4HWANG, 7 , 6)
C
 1560 CONTINUE
 1620 CONTINUE
      IF (NWING.LE.0) GO TO 1628
      DO 1625 J=1,NWING
 1625  ALPHAX(J)=ALPHAX(J)+AWS(J)
 1628  CONTINUE
      CALL FOR EOF(NTAPEC)
      REWIND NTAPEC
C
C     ANY CONFIGURATION
C
      CALL OVERLAY(4HWANG, 7, 7)
C
C        FOR GRAPHIC REPRESENTATION, WRITE ON TAPE CPBB, CPB, CPU,CPL
C
      IREPT = 1
      IF ( IPLOT .EQ. 1 ) CALL OVERLAY(4HWANG, 7, 13)
 1111 CONTINUE
      IF ( IRW  .GT. 0 ) CALL OVERLAY(4HWANG, 7, 11)
C
      IF ( KONFIG  .LT. 3 ) GO TO 1600
C
C        ADDING PRESSURE DUE TO THE PRESENCE OF WING TO BODY PANEL
C
      CALL OVERLAY(4HWANG, 7, 9)
```

```
C
      IREPT = 2
      IF ( IPLOT .EQ. 1 ) CALL OVERLAY(4HWANG, 7, 13)
C
 1800 CONTINUE
C
      NSYM = 1
C
      GO TO ( 1820, 1810, 1810 ) , KONFIG
C
C
C               INTERPOLATION OF PRESSURE
C
 1810 CONTINUE
C
C               BODY
C
      CALL OVERLAY(4HWANG,7, 8)
C
      IF ( KONFIG  .EQ.  3 ) NSYM = NSYM + 1
C
      IF ( KONFIG  .EQ. 2 ) GO TO 1840
C
 1820 CONTINUE
C
C               WING
C
      CALL OVERLAY (4HWANG, 7, 10)
 1840 CONTINUE
C
C
      NPOLAR = 1
C
C     ANY CONFIGURATION
C     READ ADDITIONAL POLARS
      IF ( POLAR ) 1630, 1680, 1630
C
 1630 CONTINUE
C
C
 1640 CONTINUE
      IPOLAR=1
      DARAD=DADEG/CDR
      ARA=ARA+DARAD
      ARADEG=ARA*CDR
      WRITE (NTAPEO,1710) TITLE
      WRITE (NTAPEO,1720) DADEG
      WRITE (NTAPEO,1760) ARADEG
C
      IF (KONFIG .EQ. 2) GO TO 1670
C
C     WING ALONE OR WING-BODY CONFIGURATION
      DO 1650 I=1,NWING
      JW=I+NBODY
      COST=COS(THETA(JW) + BROLL )
 1650 ALPHAW(I)=ALPHAW(I)-DARAD*COST
C
      IF (KONFIG .EQ. 1) GO TO 1670
C
C     WING-BODY CONFIGURATION
      DO 1660 I=1,NBODY
 1660 ALPHAB(I)=ALPHAB(I)-DARAD*COS(THETA(I))
C
C     ANY CONFIGURATION
 1670 CONTINUE
```

```
C
      IF(KONFIG-2)1460,1480,1480
C
 1680 CONTINUE
C
C---------------
C ------------------
C
C
C
 1690 FORMAT(20A4)
 1700 FORMAT(7F10.9)
 1710 FORMAT(1H1,20A4,30X,2A4)
 1720 FORMAT(1H0,23HDELTA ANGLE OF ATTACK =,F10.4,2X,4HDEG.)
 1730 FORMAT(1H0,27HWING OPTIMIZED FOR CL BAR =,F10.4)
 1740 FORMAT(1H , 18X,9HXCP BAR =,F10.4)
 1750 FORMAT(1H0,56HINCLINATION OF BODY AXIS WITH RESPECT TO DEFINING AX
     1IS =,F10.4,2X,4HDEG.)
 1760 FORMAT(1H0,43HANGLE OF ATTACK WITH RESPECT TO BODY AXIS =,F10.4,
     12X,4HDEG.)
 1765 FORMAT (1H0,*BODY YAW = *,F10.2,* DEG *)
 1766 FORMAT (,H0,*BODY ROLL = *,F10.2,* DEG *)
 1770 FORMAT(1H0,38HHEIGHT OF WING PLANE ABOVE BODY AXIS =,F10.4)
 1780 FORMAT (1H0,*BODY REFERENCE AREA  =  *,F10.4)
      RETURN
      END
C
C/    LIST,NONE
      SUBROUTINE TVEL(A,B,C,D,ALPHAT,UBWT,VBWT,WBWT,UWWT,VWWT,WWWT,AN,
     1CHORD,THICK,NROW)
C
C ......................................................................
C     COMPUTE VELOCITY COMPONENTS INDUCED BY WING SOURCES
C ......................................................................
C
      COMMON/ MAIN / NTAPEA,NTAPEB,NTAPEC,NTAPED,NTAPEE,NTAPEF,NTAPEI,
     1          NTAPEO,NBODY,NWING,XMACH,SYM,KACE,NPOLAR,IRW
      COMMON /PANVEL/ VSS(220),WS(110)
C
      INTEGER TSL
      REAL ATLE(20)
      DIMENSION A(1),B(1),C(1),D(1),ALPHAT(1),UBWT(1),VBWT(1),WBWT(1),
     1UWWT(1),VWWT(1),WWWT(1),AN(1),NROW(1),CHORD(1)
C
C
      DATA TSL /4HGIVE/
C
      NS=NBODY+1
      NPANEL=NBODY+NWING
      IF (NBODY .EQ. 0) GO TO 1010
      DO 1000 J=1,NBODY
      UBWT(J)=0.
      VBWT(J)=0.
      WBWT(J)=0.
      AN(J)=0.
 1000 CONTINUE
 1010 DO 1020 J=1,NWING
      UWWT(J)=0.
      VWWT(J)=0.
      WWWT(J)=0.
      JJ=J+NBODY
      AN(JJ)=0.0
 1020 CONTINUE
C
      GO TO (1030,1180,1040),KACE
```

```
C
 1030 NC=NWING/NROW(1)
      NR=NROW(1)
      GO TO 1050
 1040 NC=NWING/NROW(2)
      NR=NROW(2)
 1050 N=0
      NWINC=NWING+NC
      IF ( THICK  .LT. 1.0 ) GO TO 1080
C
C     INPUT THICKNESS SLOPES
      K = -NR + 1
      DO 1070 I=1,NC
       K = K+NR
       KK = K+NR - 1
      READ (NTAPEI,1190) ATLE(I),(ALPHAT(J),J=K,KK)
 1070 CONTINUE
      GO TO 1110
C
C     USE PREVIOUSLY CALCULATED PANEL SLOPES
C     AND COMPUTE LEADING EDGE SLOPES
 1080 CALL FSF(1,NTAPEC,IRR)
      READ (NTAPEC) (ALPHAT(I),I=1,NWING)
      REWIND NTAPEC
      NR1=NR-1
C
      DO 1100 J=1,NC
      J1=(J-1)*NR
      JN=NBODY+J1
      AT=-ALPHAT(J1+1)
      DO 1090 I=1,NR1
      IJ=J1+I
      IJN=NBODY+IJ
      AT=AT-(ALPHAT(IJ)+ALPHAT(IJ+1))*CHORD(IJN+1)/
     1CHORD(JN+1)
 1090 CONTINUE
      ATLE(J)=AT
 1100 CONTINUE
C
 1110 CONTINUE
      CALL FSF(8,NTAPEC,IRR)
      WRITE (NTAPEC) NWINC
      DO 1118 J=1,NC
      J1=(J-1)*NR
      DO 1116 I=1,NR
      IJ=I+J1
      IF(I.EQ.1) GO TO 1114
      WS(IJ)=(ALPHAT(IJ-1)+ALPHAT(IJ))*.5
      GO TO 1116
 1114 WS(IJ)=(ATLE(J)+ALPHAT(IJ))*.5
 1116 CONTINUE
 1118 CONTINUE
      DO 1170J=1,NWINC
      READ (NTAPEA) (D(I),I=1,NPANEL)
      READ (NTAPEB) (A(I),B(I),C(I),I=1,NPANEL)
      JT=J-N*(NR+1)
      IF (JT)1130,1130,1120
 1120 N=N+1
      WT=ATLE(N)
      GO TO 1140
 1130 JN=J-N
      WT=ALPHAT(JN)
 1140 CONTINUE
      WRITE (NTAPEC) WT
C
```

```
      DO 1160 I=1,NPANEL
      IF (I .GT. NBODY) GO TO 1150
      UBWT(I)=UBWT(I)+A(I)*WT
      VBWT(I)=VBWT(I)+B(I)*WT
      WBWT(I)=WBWT(I)+C(I)*WT
      AN(I)=AN(I)+D(I)*WT
      GO TO 1160
 1150 IN=I-NBODY
      UWWT(IN)=UWWT(IN)+A(I)*WT
      VWWT(IN)=VWWT(IN)+B(I)*WT
      WWWT(IN)=WWWT(IN)+C(I)*WT
      AN(I)=AN(I)+D(I)*WT
 1160 CONTINUE
 1170 CONTINUE
C
 1180 CONTINUE
      CALL FOR EOF (NTAPEC)
      REWIND NTAPEC
C
C ------------------
      RETURN
C ------------------
C
 1190 FORMAT (7F10.0)
      END
C
      SUBROUTINE READ(NTAPEI,NTAPEO,NTAPEC,NM,S,KACE,ICAMBR,NROW)
C
C ...................................................................
C     CONTROLS DATA READ IN OPTION FOR SPECIFYING WING CAMBER AND
C     PRESSURE DISTRIBUTION, ALSO BODY CAMBER
C ...................................................................
C
      DIMENSION NROW(2)
C
      REAL S(1)
      INTEGER DICT(2)
C
      DATA DICT(1)/4HCONS/
      DATA DICT(2)/4HGIVE/
C
C        ICOMBR = 1  CONSTANT CAMBER, 2  COMPUTED, 3  READ IN
      GO TO ( 1060, 1080, 1000), ICAMBR
 1000 CONTINUE
      IF (KACE .NE. 0) GO TO 1010
      READ (NTAPEI,1100) (S(I),I=1,NM)
      GO TO 1090
C
 1010 GO TO (1020,1090,1030), KACE
 1020 NC = NM / NROW(1)
      NR = NROW(1)
      GO TO 1040
 1030 NC = NM / NROW(2)
      NR = NROW(2)
 1040 K = -NR + 1
      DO 1050 I=1,NC
       K = K+NR
       KK = K+NR - 1
       READ (NTAPEI,1100) (S(J),J=K,KK)
 1050 CONTINUE
       GO TO 1090
C
 1060 CONTINUE
      READ (NTAPEI,1100) SCONST
      DO 1070 I=1,NM
 1070 S(I)=SCONST
      GO TO 1090
 1080 CONTINUE
      READ (NTAPEC) (S(I),I=1,NN)
 1090 CONTINUE
C
C -----------------
      RETURN
C -----------------
 1100 FORMAT(7F10.0)
      END
C
      SUBROUTINE BTHICK(NBODYS,NBODY,NROWB,XB,R,XC,ALPHA,THETA)
C
C ....................................................................
C     COMPUTES BODY THICKNESS SLOPES FOR GIVEN BODY RADII DISTRIBUTION
C ....................................................................
C
      DIMENSION XB(1),R(1),XC(1),ALPHA(1),THETA(1)
C
      NB=NBODYS-1
      NCOLB=NBODY/NROWB
C
      DO 1000 I=1,NB
 1000 ALPHA(I+1)=(R(I+1)-R(I))/(XB(I+1)-XB(I))
      ALPHA(1)=ALPHA(2)
      DO 1030 J=1,NROWB
      DO 1010 I=1,NB
      IF (XB(I)-XC(J))1010,1020,1020
 1010 CONTINUE
      . -NBODYS
 1020 THETA(J)=ALPHA(I-1)+(XC(J)-XB(I-1))/(XB(I)-XB(I-1))*(ALPHA(I)-
     1ALPHA(I-1))
 1030 CONTINUE
C
      DO 1040 J=1,NROWB
      DO 1040 I=2,NCOLB
      IJ=(I-1)*NROWB+J
      THETA(IJ)=THETA(J)
 1040 CONTINUE
C
C -----------------
      RETURN
C -----------------
      END
C
      SUBROUTINE BCAM(NBODYS,NROWB,XB,ZDELTA,XC,DZDXB,ACB)
C
C ....................................................................
C     COMPUTES BODY CAMBER SLOPES, GIVEN BODY CAMBER SHAPE
C ....................................................................
C
      DIMENSION XB(1),ZDELTA(1),XC(1),DZDXB(1),ACB(1)
C
      NB=NBODYS-1
      DO 1000 I=1,NB
 1000 DZDXB(I+1)=(ZDELTA(I+1)-ZDELTA(I))/(XB(I+1)-XB(I))
      DZDXB(1)=DZDXB(2)
C
      DO 1030 J=1,NROWB
      DO 1010 I=1,NB
      IF (XB(I)-XC(J))1010,1020,1020
 1010 CONTINUE
 1020 ACB(J)=DZDXB(I-1)+(XC(J)-XB(I-1))/(XB(I)-XB(I-1))*(DZDXB(I)-DZDXB(
     1I-1))
 1030 CONTINUE
C
C ------------------
      RETURN
C ------------------
      END
C
      SUBROUTINE INOUT(NTAPEO,KASE,CPCALC,POLAR,THICK,VOUT,XMACH,SEMIS,
     1XP,ZP,CBAR,RFAREA,SYM)
C
C ......................................................................
C     PRINTS OUT ALL INPUT AERODYNAMIC DATA
C ......................................................................
C
C
      WRITE (NTAPEO,1200)
      IF (SYM)1010,1000,1010
 1000 WRITE (NTAPEO,1210)
      GO TO 1020
 1010 WRITE (NTAPEO,1220)
 1020 CONTINUE
C
      GO TO (1030,1040,1050),KASE
 1030 WRITE (NTAPEO,1230)
      GO TO 1060
 1040 WRITE (NTAPEO,1240)
      GO TO 1060
 1050 WRITE (NTAPEO,1250)
 1060 CONTINUE
C
      IF (CPCALC-1.)1070,1080,1090
 1070 WRITE (NTAPEO,1260)
      GO TO 1100
 1080 WRITE (NTAPEO,1270)
      GO TO 1100
 1090 WRITE (NTAPEO,1280)
 1100 CONTINUE
C
      IF (POLAR 1120,1110,1120
 1110 WRITE (NTAPEO,1290)
      GO TO 1130
 1120 WRITE (NTAPEO,1300)
 1130 CONTINUE
C
      IF (THICK)1150,1140,1150
 1140 WRITE (NTAPEO,1310)
      GO TO 1160
 1150 WRITE (NTAPEO,1320)
 1160 CONTINUE
C
      IF (VOUT)1180,1170,1180
 1170 WRITE (NTAPEO,1330)
      GO TO 1190
 1180 WRITE (NTAPEO,1340)
 1190 CONTINUE
C
      WRITE (NTAPEO,1380) XMACH
      WRITE (NTAPEO,1370) XP,ZP,CBAR
      WRITE (NTAPEO,1360) RFAREA
      WRITE (NTAPEO,1350) SEMIS
C
C
C
C ------------------
      RETURN
```

```
C -------------------
 1200 FORMAT(1H0//30H DESCRIPTION OF CASE REQUESTED)
 1210 FORMAT(1H0/85H UNSYMMETRICAL CONFIGURATION - PANELS LOCATED ONLY O
     1N ONE SIDE OF X-Z PLANE(SYM = 0.))
 1220 FORMAT(1H0/80H SYMMETRICAL CONFIGURATION - PANELS LOCATED ON BOTH
     1SIDES OF X-Z PLANE(SYM = 1.))
 1230 FORMAT(1H0,9HCASE = 1.,10X,35HCALCULATE WING SHAPE, GIVEN WING CL)
 1240 FORMAT(1H0,9HCASE = 2.,10X,25HCALCULATE CL, GIVEN SHAPE)
 1250 FORMAT(1H0,9HCASE = 3.,10X,19HOPTIMIZE WING SHAPE)
 1260 FORMAT(1H0,11HCPCALC = 0.,8X,9HLINEAR CP)
 1270 FORMAT(1H0,11HCPCALC = 1.,8X,13HNON-LINEAR CP)
 1280 FORMAT(1H0,11HCPCALC = 2.,8X,8HEXACT CP)
 1290 FORMAT(1H0,10HPOLAR = 0. ,9X,20HPOLARS NOT REQUESTED)
 1300 FORMAT(1H0,10HPOLAR = 1. ,9X,16HPOLARS REQUESTED)
 1310 FORMAT(1H0,10HTHICK = 0.,9X,40HWING THICKNESS PRESSURES NOT TO BE
     1ADDED)
 1320 FORMAT(1H0,10HTHICK = 1.,9X,36HWING THICKNESS PRESSURES TO BE ADDE
     1D)
 1330 FORMAT(1H0,9HVOUT = 0.,10X,37HVELOCITY COMPONENTS NOT TO BE PRINTE
     1D)
 1340 FORMAT(1H0,9HVOUT = 1.,10X,33HVELOCITY COMPONENTS TO BE PRINTED)
 1350 FORMAT (1H0,17HWING SEMI-SPAN = ,F10.4)
 1360 FORMAT(1H0,21HWING REFERENCE AREA =,F10.4)
 1370 FORMAT(1H0,48HPOINT ABOUT WHICH THE MOMENTS ARE TO BE COMPUTED/16H
     1  X-COORDINATE =,F10.4/16H  Z-COORDINATE =,F10.4//
     225H REFERENCE CHORD LENGTH =,F10.4)
 1380 FORMAT(1H0,13HMACH NUMBER =,F10.4)
      END
C
      SUBROUTINE CENTRO
C
C
C              COMPUTES CENTROID COORDINATES OF FINITE ELEMENT,
C            TRIANGULAR OR QUADRILATERAL.
      COMMON/PEDATA/ NBODYS,XMIDB(51), NTHETB,THETB(15),CPBB(51,15)
     1         ,XC(3,700),NG(350),NELM(700),NE,NGRIDP,PINF,IPUNCH, CPB
     2         , SID(20)
      COMMON/ELMCNT/ X(3, 350)
C
      DIMENSION CPB(150)
      DATA IWORD1 / 5HCQUAD /
C
C              READ FINITE ELEMENT GRID SPECIFICATION FROM BING OUTPUTS.
C
      DO 200 I= 1, NE
      READ (5, 901) ICHECK, ID, IE, ND, NG1, NG2, NG3, F64, DM
      NELM(I) = IE
      NG4 = F64
C
C              SEARCH GRID POINT NUMBER
      ICOUNT = 1
      DO 110 K=1,NGRIDP
      IF ( NG(K)   .EQ. NG1 ) GO TO 101
      IF ( NG(K)   .EQ. NG2 ) GO TO 102
      IF ( NG(K)   .EQ. NG3 ) GO TO 103
      IF ( NG(K)   .EQ. NG4 ) GO TO 104
      GO TO 110
  101 I1 = K
      GO TO 109
  102 I2 = K
      GO TO 109
  103 I3 = K
      GO TO 109
  104 I4 = K
  109 IF ( ICOUNT  .EQ.  4 ) GO TO 150
```

```
      IF (  ICOUNT  .EQ. 3  .AND.  ICHECK  .NE.  IWORD1 ) GO TO 111
      ICOUNT = ICOUNT + 1
  110 CONTINUE
C
C              TRIANGULAR ELEMENT
C
  111 CONTINUE
      DO 120  J= 1, 3
      XC( J, I) = ( X( J, I1) + X( J, I2) + X( J, I3) ) / 3.0
  120 CONTINUE
      GO TO 200
C
  150 CONTINUE
C
C              QUADRILATERAL ELEMENT
C
      X1 = ( X( 1, I1 ) + X( 1, I2 ) + X( 1, I3 ) ) / 3.0
      Y1 = ( X( 2, I1 ) + X( 2, I2 ) + X( 2, I3 ) ) / 3.0
      Z1 = ( X( 3, I1 ) + X( 3, I2 ) + X( 3, I3 ) ) / 3.0
      X2 = ( X( 1, I3 ) + X( 1, I4 ) + X( 1, I1 ) ) / 3.0
      Y2 = ( X( 2, I3 ) + X( 2, I4 ) + X( 2, I1 ) ) / 3.0
      Z2 = ( X( 3, I3 ) + X( 3, I4 ) + X( 3, I1 ) ) / 3.0
C
      S12 = SQRT( ( X(1,I1) - X(1,I2) )**2 + ( X(2,I1) - X(2,I2) )**2
     1             + ( X(3,I1) - X(3,I2) )**2 )
      S23 = SQRT( ( X(1,I2) - X(1,I3) )**2 + ( X(2,I2) - X(2,I3) )**2
     1             + ( X(3,I2) - X(3,I3) )**2 )
      S31 = SQRT( ( X(1,I3) - X(1,I1) )**2 + ( X(2,I3) - X(2,I1) )**2
     1             + ( X(3,I3) - X(3,I1) )**2 )
      S34 = SQRT( ( X(1,I3) - X(1,I4) )**2 + ( X(2,I3) - X(2,I4) )**2
     1             + ( X(3,I3) - X(3,I4) )**2 )
      S41 = SQRT( ( X(1,I4) - X(1,I1) )**2 + ( X(2,I4) - X(2,I1) )**2
     1             + ( X(3,I4) - X(3,I1) )**2 )
      ST123 = 0.5 * ( S12 + S23 + S31 )
      ST341 = 0.5 * ( S31 + S34 + S41 )
      A123 = SQRT( ST123 * ( ST123-S12) * ( ST123-S23) * (ST123-S31) )
      A341 = SQRT( ST341*( ST341-S31)*(ST341-S34)*(ST341-S41) )
      AT = A123 + A341
      XC(1,I) = ( X1*A123 + X2*A341 ) / AT
      XC(2,I) = ( Y1*A123 + Y2*A341 ) / AT
      XC(3,I) = ( Z1*A123 + Z2*A341 ) / AT
C
  200 CONTINUE
C
  901 FORMAT ( A5, A3, 5I8, F8.0, F8.4 )
C
      RETURN
      END
C
      SUBROUTINE CONVT( X, NG, NGRIDP, NSYM, ISYM)
C
C
C              CONVERT GRID POINT COORDINATES TO THE RECTANGULAR COORD.
C
      DIMENSION X( 3, NGRIDP), NG(NGRIDP), ISYM(20)
C
      DR = 0.0174532925
C
      DO 250  I = 1, NGRIDP
C
C              READ OUTPUT OF DING  FINITE ELEMENT GRID POINT COORDINATE
C
      READ (5, 901) CHECK,IDUM,NG(I), NS, X1, X2, X3, ND
      R = X1
      THETA = X2
      PHI = X3
      NGO = ISYM(NSYM) + 1
      GO TO ( 100, 150, 200 ) , NGO
C
C              RECTANGULAR COORD.
C
  100 CONTINUE
      X( 1, I) = X1
      X( 2, I) = X2
      X( 3, I) = X3
      GO TO 250
C              CYLINDRICAL COORD.
C
  150 CONTINUE
      X( 1, I) = R * COS( THETA * DR )
      X( 2, I) = R * SIN( THETA * DR )
      X( 3, I) = X3
      GO TO 250
C
C              SPHERICAL COORD.
C
  200 CONTINUE
      X( 1, I) = R * SIN( THETA*DR ) * COS( PHI * DR )
      X( 2, I) = R * SIN( THETA*DR ) * SIN( PHI * DR )
      X( 3, I) = R * COS( THETA*DR )
C
  250 CONTINUE
C
  901 FORMAT ( A5,A3, 2I8, 3F8.2, I8 )
C
      RETURN
      END
C
      SUBROUTINE COORDT( X, NG)
C
C
C              TRANSFORM COORDINATES OF THE GRID POINT OF FINITE ELEMENT
C              TO THE AERO CODE SYSTEM USING THE TRANSFORM MATRIX TFM
C
      COMMON/CORTRN/ NSYM,NTRANS(20),ISYM(20),VA(3,2,20), BTFM(9,2,20)
      DIMENSION X(3,NG), TX(3)
C
      NT = NTRANS(NSYM)
      DO 300 I=1, NG
C
      DO 210 ITRAN = 1, NT
C
      DO 200 J=1,3
C
      SUM = 0.0
      DO 150 K=1, 3
      JK = K + ( J-1) * 3
      SUM = SUM + X( K,I) * BTFM( JK, ITRAN, NSYM)
  150 CONTINUE
      TX(J) = SUM + VA( J, ITRAN, NSYM)
  200 CONTINUE
      DO 205  M=1,3
      X( M, I) = TX( M )
  205 CONTINUE
C
  210 CONTINUE
C
  300 CONTINUE
C
      RETURN
      END
C
      SUBROUTINE SURFIT( X, Y, N, B, TM)
C
C        COMPUTES COEFFICIENTS FOR SURFACE SPLINE.
C        N  TOTAL NUMBER OF INPUT POINTS, CAN BE RANDOMLY LOCATED.
C        JEQ  FOR EQUATION NUMBERING
C        MUK  FOR COEFFICIENT NUMBERING
C
C        CF  ARE THE COEFFICINT MATRIX OF EQUATION
C              W(X,Y) = B(1) + B(2)*X + B(3)*Y + SUM OF B(I) * R(I)**2 *
C              LN( R(I)**2)
C
      COMMON/REPEAT/ IREPT
      COMMON/MATRIX/ CF(115,115)
      DIMENSION IPS(115), TM(1), SCALES(115)
      DIMENSION X( 1), Y( 1), B( 1)
C
      N6 = 115
      N1 = N + 1
      N2 = N + 2
      N3 = N + 3
C
C        BYPASS MATRIX INVERSION FOR THE CASE WHERE THE DATA POINTS (X,
C        Y )  ARE IDENTICAL FOR INTERPOLATION
C
C
      IF ( IREPT .GT. 1 ) GO TO 300
      DO 100 JEQ = 1, N
      CF( JEQ, 1) = 1.0
      CF ( JEQ, 2 ) =   X( JEQ )
      CF ( JEQ, 3 ) =   Y( JEQ )
      DO 90 MUK = 1, N
      MUK1 = MUK + 3
      IF ( MUK  .EQ.  JEQ ) GO TO 85
      RJM = ( X( MUK ) - X( JEQ ) )**2 + ( Y( MUK ) - Y( JEQ ) )**2
      CF( JEQ, MUK1 ) = RJM * ALOG( RJM )
      GO TO 90
   85 CF( JEQ, MUK1 ) = 0.0
   90 CONTINUE
  100 CONTINUE
C
      DO 150 K=1,3
      CF( N1, K) = 0.0
      CF( N2, K) = 0.0
      CF( N3, K) = 0.0
  150 CONTINUE
      DO 200  MUK =1,N
      MUK1 = MUK + 3
      CF( N1, MUK1) = 1.0
      CF ( N2, MUK1) =   X( MUK )
      CF( N3, MUK1 ) =   Y( MUK )
  200 CONTINUE
C
C
C
  250 CONTINUE
C
      CALL DECOMP( N3, CF, IPS, SCALES)
C
  300 CONTINUE
      B(N1) = B(N2) = B(N3) = 0.0
      CALL SOLVE( N3, CF, B, TM,IPS)
C
C
```

```
      RETURN
      END
      SUBROUTINE DECOMP( NN, A, IPS, SCALES)
C
C
C        DECOMPOSING OF MATRIX A
C
      DIMENSION A(115, 115), SCALES( 1), IPS( 1)
C
      N = NN
C
C        INITIALIZE IPS, A AND SCALES
C
      DO 5 I = 1, N
      IPS(I) = I
      ROWNRM = 0.0
      DO 2 J = 1, N
      IF ( ROWNRM - ABS( A( I,J) ) ) 1, 2, 2
    1 ROWNRM = ABS( A( I, J) )
    2 CONTINUE
      IF ( ROWNRM ) 3, 4, 3
    3 SCALES(I) = 1.0 / ROWNRM
      GO TO 5
    4 CALL SING(1)
      SCALES(I) = 0.0
    5 CONTINUE
C
C        GAUSSIAN ELIMINATION WITH PARTIAL PIVOTING
C
      NM1 = N - 1
      DO 17 K = 1, NM1
      BIG = 0.0
      DO 11 I = K, N
      IP = IPS(I)
      SIZE = ABS( A( IP, K) ) * SCALES(IP)
      IF ( SIZE - BIG ) 11, 11, 10
   10 BIG = SIZE
      IDXPIV = I
   11 CONTINUE
      IF ( BIG) 13, 12, 13
   12 CALL SING(2)
      GO TO 17
   13 IF ( IDXPIV - K) 14, 15, 14
   14 J = IPS(K)
      IPS(K) = IPS( IDXPIV )
      IPS(IDXPIV) = J
   15 KP = IPS(K)
      PIVOT = A( KP, K)
      KP1 = K + 1
      DO 16 I = KP1, N
      IP = IPS(I)
      EM = - A( IP, K) / PIVOT
      A( IP, K) = - EM
      DO 16 J = KP1, N
      A( IP, J) = A( IP, J ) + EM * A( KP, J )
C
C        INNER LOOP
C
   16 CONTINUE
   17 CONTINUE
      KP = IPS(N)
      IF ( A( KP, N ) ) 19, 18, 19
   18 CALL SING(2)
   19 CONTINUE
C
```

```
      RETURN
      END
      SUBROUTINE SOLVE( NN, A, B, X, IPS)
C
C        SOLVE FOR MATRIX  A  *  X  =  B
C
C
      DIMENSION A(115, 115), B( 1), X( 1), IPS(1)
C
      N = NN
      NP1 = N + 1
C
      IP = IPS(1)
      X(1) = B(IP)
      DO 2 I=2,N
      IP = IPS(I)
      IM1 = I - 1
      SUM = 0.0
      DO 1 J = 1, IM1
    1 SUM = SUM + A(IP, J) * X(J)
    2 X(I) = B(IP) - SUM
C
      IP = IPS(N)
      X(N)=X(N) / A( IP, N)
      DO 4 IBACK = 2, N
C
C        I  GOES FROM  (N-1) . . . . . 1
C
      I = NP1 - IBACK
      IP = IPS(I)
      IP1 = I + 1
      SUM = 0.0
      DO 3 J = IP1, N
    3 SUM = SUM + A( IP, J) * X(J)
    4 X(I)=( X(I) - SUM ) / A( IP, I)
C
C
      RETURN
      END
      SUBROUTINE SING(IWHY)
C
C        PRINTS STATEMENT
C
C        NOUT = STANDARD OUTPUT UNIT
C
      NOUT = 6
      GO TO ( 1, 2, 3), IWHY
    1 WRITE ( NOUT, 11)
      GO TO 10
    2 WRITE ( NOUT, 12)
      GO TO 10
    3 WRITE ( NOUT, 13)
C
   11 FORMAT (1HO,*MATRIX WITH ZERO ROW IN DECOMPOSE*)
   12 FORMAT (1HO,*SINGULAR MATRIX IN DECOMPOSE, ZERO DIVIDE IN SOLVE*)
   13 FORMAT (1HO,*NO CONVERGENCE IN IMPRUV. MATRIX NEARLY SINGULAR*)
C
   10 CONTINUE
C
      RETURN
      END
C
      SUBROUTINE INVERT (GOTON, IA, IPIVOT, INDXR, INDXC, NROWS)
C
C ..................................................................
```

```
C
C .................................................................
C
      COMMON/MATRIX/ A(115,115)
      INTEGER IPIVOT(IA), INDXR(IA), INDXC(IA)
C
C
      GOTON = 1
      N = IA
      DO 1000 J=1, N
 1000 IPIVOT(J) = 0
      DO 1090 I=1, N
      T = 0.0
      DO 1020 J=1, N
      IF (IPIVOT(J).EQ.1) GO TO 1020
C
      DO 1010 K=1, N
      IF (IPIVOT(K).EQ.1) GO TO 1010
      IF ( .NOT. (ABS(A(J,K)) -ABS(T) .GT. 0.0) ) GO TO 1010
      IROW = J
      ICOL = K
      T = A(J,K)
 1010 CONTINUE
 1020 CONTINUE
      IPIVOT(ICOL) = IPIVOT(ICOL)+1
      IF (IROW.EQ.ICOL) GO TO 1040
      DO 1030 L=1, N
      T = A(IROW,L)
      A(IROW,L) = A(ICOL,L)
 1030 A(ICOL,L) = T
 1040 INDXR(I) = IROW
      INDXC(I) = ICOL
      PIVOT = A(ICOL,ICOL)
      IF (PIVOT)1050,1120,1050
 1050 A(ICOL,ICOL) = 1.0
      DO 1060 L=1, N
 1060 A(ICOL,L) = A(ICOL,L)/PIVOT
      DO 1080 L=1, N
      IF (L.EQ.ICOL) GO TO 1080
      T = A(L,ICOL)
      A(L,ICOL) = 0.0
      DO 1070 M=1, N
 1070 A(L,M) = A(L,M)-A(ICOL,M)*T
 1080 CONTINUE
 1090 CONTINUE
      DO 1110 I=1, N
      L = N+1-I
      IF( INDXR(L)  .EQ.  INDXC(L) )  GO TO 1110
      IROW = INDXR(L)
      ICOL = INDXC(L)
      DO 1100 K=1, N
      T = A(K,IROW)
      A(K,IROW) = A(K,ICOL)
 1100 A(K,ICOL) = T
 1110 CONTINUE
C
C
C     SUCCESSFUL SOLUTION
C
C ------------------
      RETURN
C ------------------
C
 1120 CONTINUE
C
```

```
C     SINGULAR MATRIX
      WRITE (6,1130)
C
C ------------------
      GOTON = 2
      RETURN
C ------------------
 1130 FORMAT (29H ERROR THE MATRIX IS SINGULAR)
      END
C
      SUBROUTINE SINVRT ( NROWS, N, IR1, IR2, SCALE, DET, NDETXP)
C
C ................................................................
C
C ................................................................
C
      COMMON/MATRIX/ C1(115,115)
      INTEGER N1(115), N2(115), N3(115)
C
      IR1 = 0
      IR2 =0
      SCALE = 0.0
      DET = 0.0
      NDETXP = 0.0
      CALL INVERT(GOTON,N,N1,N2,N3,NROWS)
      IF (GOTON.EQ.2) GO TO 1000
C
C ------------------
      RETURN
C ------------------
 1000 CONTINUE
      IR1 =2
C
C ------------------
      RETURN'
C ------------------
      END
C
      SUBROUTINE SIMP(R,X,Y,N,IER)
      DIMENSION X(N),Y(N)
      R=0.0
      IF(N.GT.1) GO TO 1000
      IER=2
      RETURN
 1000 IF(X(1).EQ.X(2)) GO TO 1110
      NM1=N-1
      IF(N.EQ.2) GO TO 1120
      IF(X(1).LT.X(2)) GO TO 1020
C  TEST FOR X TO BE MONOTONICALLY DECREASING
      DO 1010 I=2,NM1
      IF(X(I+1).GE.X(I)) GO TO 1110
 1010 CONTINUE
      GO TO 1040
C  TEST FOR X TO BE MONOTONICALLY INCREASING
 1020 DO 1030 I=2,NM1
      IF(X(I+1).LE.X(I)) GO TO 1110
 1030 CONTINUE
 1040 NM2=N-2
      IF(MOD(N,2).EQ.0) GO TO 1130
      P=0.0
      N1=1
 1050 S1=X(N1+1)-X(N1)
      S2=X(N1+2)-X(N1+1)
      S3=X(NM1)-X(NM2)
      S4=X(N)-X(NM1)
      R=(2.*S1**2+S1*S2-S2**2)/S1*Y(N1)+(2.*S4**2+S3*S4-S3**2)/S4*Y(N)
      N1=N1+1
      DO 1060 I=N1,NM1,2
      S1=X(I)-X(I-1)
      S2=X(I+1)-X(I)
 1060 R=R+(S1+S2)**3/(S1*S2)*Y(I)
      IF(N.LT.5) GO TO 1080
      N1=N1+1
      DO 1070 I=N1,NM2,2
      S1=X(I-1)-X(I-2)
      S2=X(I)-X(I-1)
      S3=X(I+1)-X(I)
      S4=X(I+2)-X(I+1)
 1070 R=R+((2.*S2**2+S1*S2-S1**2)/S2+(2.*S3**2+S3*S4-S4**2)/S3)*Y(I)
 1080 R=R/6.+P
 1090 K = LEGVAR(R)
      IF (K.NE.0) GO TO 1100
      IF(K.EQ.1) GO TO 1100
      IER = 1
      RETURN
 1100 IER=3
      RETURN
 1110 IER=4
      RETURN
C  TRAPEZOIDAL RULE FOR N=2
 1120 R=(X(2)-X(1))*(Y(1)+Y(2))/2.0
      GO TO 1090
C  FIT POLYNOMIAL THRU FIRST 3 POINTS AND INTEGRATE FROM X(1) TO X(2).
 1130 S1=X(2)-X(1)
      S2=X(3)-X(1)
      S3=Y(2)-Y(1)
      S4=Y(3)-Y(1)
      P=S1/6.*(2.*S3+6.*Y(1)+(S2**2*S3-S1**2*S4)/(S2*(S2-S1)))
      N1=2
      GO TO 1050
      END
      OVERLAY(WANG, 7, 1)
      PROGRAM INVW
C
C ******************************************************************
C     FOR WING ONLY CASE, STORES MATRIX (A) AND (A) INVERSE ON TAPE
C     MAXIMUM SIZE MATRIX INVERSION =110
C ******************************************************************
C
      COMMON/ MAIN / NTAPEA,NTAPEB,NTAPEC,NTAPED,NTAPEE,NTAPEF,NTAPEI,
     1          NTAPEO,NBODY,NWING,XMACH,SYM,KACE,NPOLAR,IRW
C
      COMMON/MATRIX/ AWW(115,115)
C
      NDEMEN=115
C
C     READ (A) MATRIX INTO CORE, WRITE ON TAPE
      CALL FSF(1,NTAPEA,IRR)
      DO 1000 J=1,NWING
      READ (NTAPEA) (AWW(I,J),I=1,NWING)
      WRITE (NTAPEE) (AWW(I,J),I=1,NWING)
 1000 CONTINUE
      CALL FOR EOF (NTAPEE)
      REWIND NTAPEA
C
C     INVERT MATRIX (A)
      CALL SINVRT(NDEMEN,NWING,IRR1,IRR2,SCALE,DET,NDETXP)
      IF (IRR1)1010,1020,1010
 1010 WRITE (NTAPEO,1040) IRR1,IRR2,SCALE
      REWIND NTAPEE
```

```
C
C ****************
      STOP
C ****************
C
 1020 CONTINUE
      DO 1030 J=1,NWING
      WRITE (NTAPEE) (AWW(I,J),I=1,NWING)
 1030 CONTINUE
      CALL FOR EOF (NTAPEE)
      REWIND NTAPEE
C
C
C ------------------
C ------------------
C
 1040 FORMAT(1H1,38HERROR IN INVERSION OF WING ONLY MATRIX
     1,5X,6HIRR1 =,I3,5X,6HIRR2 =,I3,5X,7HSCALE =,E12.6)
      END
C
      OVERLAY(WANG, 7, 2)
      PROGRAM INVBB
C
C ..................................................................
C     FOR WING BODY CASE, PREPARES AERODYNAMIC INFLUENCE MATRIX FOR
C     MATRIX REDUCTION
C     MAXIMUM SIZE MATRIX INVERSION =110
C ..................................................................
C
      COMMON/ MAIN / NTAPEA,NTAPEB,NTAPEC,NTAPED,NTAPEE,NTAPEF,NTAPEI,
     1          NTAPEO,NBODY,NWING,XMACH,STN,K4CE,NPOLAR,IR"
C
      COMMON/MATRIX/ ABB(115,115)
      DIMENSION AWB(115)
C
C
      NDEMEN=115
C
C     READ AERODYNAMIC INFLUENCE MATRIX (INFLUENCE ON BODY DUE TO BODY
C     AND INFLUENCE ON WING DUE TO BODY) INTO CORE
C
      CALL FSF(1,NTAPEA,IRR)
      DO 1000 J=1,NBODY
      READ (NTAPEA) (ABB(I,J),I=1,NBODY),(AWB(I),I=1,NWING)
      IF ( EOF(NTAPEA) ) 1040, 990
  990 CONTINUE
C
      WRITE (NTAPED) (AWB(I),I=1,NWING)
 1000 CONTINUE
      CALL FOR EOF (NTAPED)
C
C     INVERT AERODYNAMIC INFLUENCE MATRIX (INFLUENCE ON BODY DUE TO BODY
      CALL SINVRT(NDEMEN,NBODY,IRR1,IRR2,SCALE,DET,NDETXP)
      IF (IRR1)1010,1020,1010
 1010 CONTINUE
      WRITE (NTAPEO,1050) IRR1,IRR2,SCALE
      REWIND NTAPEA
C
C ******************
      STOP
C ******************
C
 1020 CONTINUE
      DO 1030 J=1,NBODY
      WRITE (NTAPED) (ABB(I,J),I=1,NBODY)
 1030 CONTINUE
```

```
      CALL FOR EOF (NTAPED)
      REWIND NTAPED
C
C ------------------
C          PROGRAM  EXITS
      GO TO 1200
C
C ------------------
C
C     END OF FILE ON NTAPEA
 1040 WRITE (NTAPEO,1060) NTAPEA,J,NBODY,NWING
C ******************
      STOP
C ******************
 1200 CONTINUE
C
C
 1050 FORMAT(1H1,33HERROR IN INVERSION OF BODY MATRIX,5X,
     16HIRR1 =,I3,5X,6HIRR2 =,I3,5X,7HSCALE =,E12.6)
 1060 FORMAT (1H0,33H****END OF FILE DETECTED ON UNIT  ,I2,3H J= ,I5,
     1    8H  NBODY= ,I3,8H  NWING= ,I3)
      END
C
      OVERLAY(WANG, 7, 3)
      PROGRAM INVRW
C
C ***********************************************************************
C     INVERT REDUCED MATRIX
C     MAXIMUM SIZE MATRIX INVERSION =110
C ***********************************************************************
C
      COMMON/ MAIN / NTAPEA,NTAPEB,NTAPEC,NTAPED,NTAPEE,NTAPEF,NTAPEI,
     1          NTAPEO,NBODY,NWING,XMACH,S M,KACE,NPOLAR,IRW
C
      COMMON/MATRIX/ ARWW(115,115)
C
      NDEMEN=115
      DO 1000 J=1,NWING
      READ (NTAPEE) (ARWW(I,J),I=1,NW NG)
 1000 CONTINUE
      CALL FOR EOF (NTAPEE)
C
C     MATRIX INVERSION
      CALL SINVRT(NDEMEN,NWING,IRR1,IRR2,SCALE,DET,NDETXP)
      IF (IRR1)1010,1020,1010
 1010 CONTINUE
      WRITE (NTAPEO,1040) IRR1,IRR2,SCALE
      REWIND NTAPEE
C
C ******************
      STOP
C ******************
C
 1020 CONTINUE
      DO 1030 J=1,NWING
      WRITE (NTAPEE) (ARWW(I,J),I=1,NWING)
 1030 CONTINUE
      CALL FOR EOF (NTAPEE)
      REWIND NTAPEE
C
C
C ------------------
C ------------------
 1040 FORMAT(1H1,30HERROR IN INVERSION OF REDUCED MATRIX,5X,
     16HIRR1 =,I3,5X,6HIRR2 =,I3,5X,7HSCALE =,E12.6)
```

```
      END
C
      OVERLAY(WANG, 7, 4)
      PROGRAM MDMATE
C
C ......................................................................
C     MAXIMUM SIZE MATRIX INVERSION =112
C     MUST BE 2 PLUS MAXIMUM NUMBER OF WING PANELS(110)
C ......................................................................
C
C
      COMMON/ MAIN / NTAPEA,NTAPEB,NTAPEC,NTAPED,NTAPEE,NTAPEF,NTAPEI,
     1         NTAPEO,NBODY,NWING,XMACH,STM,KACE,NPOLAR,IRW
      COMMON/MATRIX/ WW(115,115)
C
      DIMENSION XBAR(210),AREA(210)
C
      NN=NWING+1
      NNN=NWING+2
C
C        DRAG MINIMIZATION OPTION ELIMINATED BY SETTING DMIN = 0.0 HERE
C        CAN BE RETORED BY PLACE THE STATEMENT " READ (LI,1020) DMIN  "
C        HERE AND REMOVE DMIN = 0.0.
C
      DMIN = 0.0
C
      IF(DMIN)1000,1000,1020
 1000 CALL FSF(1,NTAPEA,IRR)
      DO 1010 I=1,NNN
      DO 1010 J=1,NNN
 1010 WW(I,J)=0.
      GO TO 1130
 1020 MDEMEN=115
      READ (NTAPEC) NBODY,NWING,XMACH,STM,KACE
      NPANEL=NBODY+NWING
      READ (NTAPEC) (J, XBAR(I), DM1, DM2, DM3, DM4, DM5
     1,AREA(I),DUMMY6,DUMMY7,DUMMY8,I=1,NPANEL)
      REWIND NTAPEC
      DO 1030 J=1,NWING
      READ (NTAPEE) (WW(I,J),I=1,NWING)
 1030 CONTINUE
      REWIND NTAPEE
      DO 1040 I=1,NWING
      K=I
      II=I+NBODY
      DO 1040 J=1,K
      JJ=J+NBODY
      WW(I,J)=-WW(I,J)*AREA(II)-WW(J,I)*AREA(JJ)
      WW(J,I)=WW(I,J)
 1040 CONTINUE
      DO 1050 J=1,NWING
      JJ=J+NBODY
      WW(J,NN)=-AREA(JJ)
      WW(NN,J)=-AREA(JJ)
      XA=-XBAR(JJ)*AREA(JJ)
      WW(J,NNN)=XA
      WW(NNN,J)=XA
 1050 CONTINUE
      WW(NN,NN)=0.0
      WW(NN,NNN)=0.0
      WW(NNN,NN)=0.0
      WW(NNN,NNN)=0.0
      CALL FSF(1,NTAPEA,IRR)
      DO 1060 J=1,NNN
 1060 WRITE (NTAPEA)(WW(I,J),I=1,NNN)
```

```
      CALL FOR EOF (NTAPEA)
C
C     INVERT DRAG MINIMIZATION MATRIX CONSTRAINED FOR CL
      CALL SINVRT(MDEMEN,NN,IRR1,IRR2,SCALE,DET,NDETXP)
      IF (IRR1)1070,1080,1070
 1070 CONTINUE
      WRITE (NTAPEO,1170) IRR1,IRR2,SCALE
      REWIND NTAPEA
C
C ****************
      STOP
C ****************
C
 1080 CONTINUE
      DO 1090 J=1,NN
 1090 WRITE (NTAPEA)(WW(I,J),I=1,NN)
      CALL FOR EOF (NTAPEA)
      REWIND NTAPEA
      CALL FSF(1,NTAPEA,IRR)
      DO 1100 J=1,NNN
 1100 READ (NTAPEA)(WW(I,J),I=1,NNN)
      CALL FSF(2,NTAPEA,IRR)
C
C     INVERT DRAG MINIMIZATION MATRIX CONSTRAINED FOR CL AND CM
      CALL SINVRT(MDEMEN,NNN,IRR1,IRR2,SCALE,DET,NDETXP)
      IF (IRR1)1070,1110,1070
 1110 CONTINUE
      DO 1120 J=1,NNN
 1120 WRITE (NTAPEA)(WW(I,J),I=1,NNN)
      CALL FOR EOF (NTAPEA)
      REWIND NTAPEA
C
C ------------------
C          PROGRAM  EXITS
      GO TO 1200
C
C ------------------
C
 1130 DO 1140 J=1,NNN
 1140 WRITE (NTAPEA)(WW(I,J),I=1,NNN)
      CALL FOR EOF (NTAPEA)
      DO 1150 J=1,NN
 1150 WRITE (NTAPEA)(WW(I,J),I=1,NN)
      CALL FOR EOF (NTAPEA)
      GO TO 1110
C
 1200 CONTINUE
C
C     FORMATION OF DRAG MINIMIZATION MATRIX
 1160 FORMAT(7F10.0)
 1170 FORMAT(1H1,46HERROR IN INVERSION OF DRAG MINIMIZATION MATRIX
     1,6HIRR1 =,I3,5X,6HIRR2 =,I3,5X,7HSCALE =,E12.6)
      END
C
      OVERLAY(WANG, 7, 5)
      PROGRAM WANG75
C
C
      COMMON/ MAIN / NTAPEA,NTAPEB,NTAPEC,NTAPED,NTAPEE,NTAPEF,NTAPEI,
     1             NTAPEO,NBODY,NWING,XMACH,SYM,KACE,NPOLAR,IRW
      COMMON / BODYSP/ DRDX(51), DZDX(51), DRDXC(51)
      COMMON /AVAR/  A(210),ACB(21),ABX(100),AWX(110),AREA(210),ARW(2),A
     1               RWT(20),ALPHA(210),ALPHAB(210),ALPHAT(110),
     2               ALPHAW(110),ALPHAX(110),AWS(110),ABT(100),ANIN(
     3               210,2),ALPHAA,ALPHAD,ARA,ARADEG,ARB,ARW(10), ARAS,
     6              AT, AAA
      COMMON /BODS/ NUMBOD,XNOS(3),YNOS(3),ZNOS(3),NBSTAT(3)
      COMMON /BVAR/  B(210), BBCL, BBCD, BBCY, BBCM, BBCMR, BBCMY
      COMMON/CVAR/C(210), CHORD(210),CL(210),CPNN(1,1,1)
     1              ,CLS(210)
     2              ,CDR,CASE,CPCALC,CBAR,CONSNT,CLBAR,CLX,CLM,CDN
      COMMON /KVAR/  KASE,KONFIG,KPOLAR
      COMMON /NVAR/  NFMT(9),NROW(2)
     1         ,NS,NPANEL,NACEL,NROWB,NROWW,NCOLW,NTHETA,NTHETS,
     2         NXLE, NRG, NPOLR, NCLX
      COMMON/RVAR/ R(51), RN(51,2), RATIOX, RFAREA
      COMMON/TVAR/ TITLE(20), THETA(210), THETAB(15),THETAS(15), THETAN
     1         (11,2), TCL(11), TCD(11), TCM(11), THICK,TWIST
      COMMON /UVAR/  UBNT(150),UWBT(110),UWWT(110),UNCL(1,1)
      COMMON /VVAR/  VBNT(150),VWBT(110),VWWT(110),VNCL(1,1)
     1              ,VOUT
      COMMON /WVAR/  WBNT(150),WWBT(110),WWWT(110),WNCL(1,1)
      COMMON/XVAR/ XBAR(210), XC(210), XNI(2), XNXN(2), XNTN(2), XB(51),
     1         XN(51,2), XNNCD(2), XNNCL(2), XNNCM(2), XYZ(3), XCL(11),
     2         XCD(11), XNACEL, XP, XCPBAR
      COMMON /YVAR/  YBAR(210),YC(210),YNI(2)
      COMMON/ZVAR/ ZBAR(210), ZC(210), ZNI(2), ZDELTA(51), ZDN(51,2)
     1              ,ZP,ZA
      COMMON/PBDATA/ NBODYS, XNIDB(51), NTHETB, THETB(15), CPBB(51,15)
     1         ,XCB(3,700),NG(350),NELM(700),NE,NGRIDP,PINF,IPUNCH
      COMMON/EXTRA/ ARAN, DSB1, DSB2, DS1, DS2, NB1, NB2, NS1, NS2,
     1         NTN, NXN, XNOSE
      COMMON/NSLOPE/ DRNDX(51,2), DZNDX(51,2)
      COMMON/MATRIX/ VD(115,115)
      COMMON/WINGOT/ CPU, CPL,NWG,ISOLID,IFORM(10),YAW,BROLL
      DIMENSION NS1(2), NS2(2), DS1(2), DS2(2), CPU(110), CPL(110)
C
C
      IF ( KONFIG  .EQ. 1 ) GO TO 50
      CALL SUBBOD(NBODYS, NTHETB, 0, XNOSE, 0.0, 0.0, CPCALC, VOUT,ARA,
     1         XP, ZP, RFAREA, XB, R, THETB, ZDELTA, XC, YC, ZC, THETA,
     2         UWBT, VWBT, WWBT, ALPHAX, CPBB, NB1, NB2, DSB1, DSB2)
C
   50 CONTINUE
      IF ( XNACEL ) 100, 500, 100
  100 CONTINUE
C
      DO 200 N=1, NACEL
C
      DO 150 J=1,NXN
      DRDX(J) = DRNDX(J,N)
      DZDX(J) = DZNDX(J,N)
  150 CONTINUE
C
      ARAN = ARA + ARN(N) / CDR
      CALL SUBBOD( NXN, NTN, NACEL, XNI(N), YNI(N), ZNI(N), CPCALC, VOUT
     1         , ARAN, XP, ZP, RFAREA, XN(1,N), RN(1,N), THETAN(1,N),
     2         ZDN(1,N), XC, YC, ZC, THETA, UNCL(1,N), VNCL(1,N),
     3         WNCL(1,N), ANIN(1,N), CPNN(1,1,N), NS1(N), NS2(N),
     4         DS1(N), DS2(N) )
  200 CONTINUE
C
  500 CONTINUE
C
      END
C
      SUBROUTINE TRAP(XT,YT,ZT,NT)
      DIMENSION XT(1),YT(1)
      SUM=0.
      DO 1000 I=2,NT
```

```
 1000 SUM=SUM+.5*(XT(I)-XT(I-1))*(YT(I)+YT(I-1))
      ZT=SUM
      RETURN
      END
C
      SUBROUTINE SUBBOD(NBS,NT,NSTOR,XN,YN,ZN,CPC,VEL,ARA,XP,ZP,RFAREA,
     1          XB,R,THETB,ZCAM,XC,YC,ZC,TA,U,V,W,EN,CPB,NS1,NS2,DS1,DS2)
C
C     COMPUTES SINGULARITY STRENGTHS,VELOCITY COMPONENTS AND PRESSURES
C ......................................................................
C
C ......................................................................
C
      COMMON/ MAIN / NTAPEA,NTAPEB,NTAPEC,NTAPED,NTAPEE,NTAPEF,NTAPEI,
     1          NTAPEO,NBOD ,NW   ,XMACH,STM,KACE,NPOLAR,IRW
      COMMON / BODYSP/ DRDX(51), DZDX(51), DRDXC(51)
      COMMON/MATRIX/ VD(115,115)
      COMMON/WINGOT/ CPU, CPL,NWG,ISOLID,IFORM(10),YAW,BROLL
      COMMON /RVAR/ B1(210), BBCL, BBCD, BBCB, BBCM, BBCMR, BBCMY
      COMMON/INPUT / TX(2,  50)
      COMMON/ POINTP/ TY(4), TYP(4), TYPP(4)
      COMMON/ POINTS/ NPTS, NVAR
C
      DIMENSION XB(1),R(1),THETB(1),ZCAM(1),XC(1),YC(1),
     1          ZC(1), TA(1), U(1), V(1), W(1), EN(1), CPB(51,1), QC(51),
     2          THETD(22), RR(210), SY(2), RB(51), UB(22), VB(22),
     3          WB(22), UC(51), VC(51),RB2(51)
     4          ,X(51),ARC(51),Q(51),ZB(51)
      DIMENSION A(52),B(52),C(52),XX(52),YA(22),YX(22),TD(22)
      DIMENSION  CP(22),G(52), COST(22), SINT(22), CPU(110), CPL(110)
C
      EQUIVALENCE ( UC(1), RR(1)), ( VC(1), RR(52))
C
C
C     CALCULATION OF SINGULARITY STRENGTHS
C
      NP=NW+NBOD
      NB=NBS-1
      KA=0
      PI=3.1415926
      NYAW = 0
      NWPW = NW / NWG
      NT1 = NT + 1
      NT2 = 2*(NT - 1)
      NNT = NT
      NT3 = NT
      EM = XMACH
      B2 = 1.0 - EM**2
      SINA = SIN(ARA)
      COSA = COS(ARA)
      TANA = TAN(ARA)
      SINY = SIN(YAW)
      COSY = COS(YAW)
      TANY = TAN(YAW)
      A1 = 1.0/SQRT( 1.0 + TANA**2 + TANY**2 )
      COSB = A1*SQRT( 1.0 + TANA**2 )
      SINB = SQRT( 1.0 - COSB**2 )
      ARAYAW = ACOS( A1 )
      SINAY = SIN(ARAYAW)
      COSG = COS(BROLL)
      SING = SIN(BROLL)
C
      IF ( YAW .LT. 0.01 .AND. BROLL .LT. 0.01) GO TO 100
      NNT = NT2
      NT3 = NT2 + 1
      IF ( ARA .LT. 0.01 .AND. YAW .GT. 0.01) GO TO 90
      FROLL = ATAN( TAN(YAW) / TAN( ARA ) )
      GO TO 150
   90 CONTINUE
      FROLL = 0.5 * PI
      GO TO 150
  100 CONTINUE
      FROLL = 0.0
  150 CONTINUE
      DO 1000 N=1,NB
      Q(N)=0.
 1000 QC(N)=0.
      X1=XB(1)
      DO 1010 N=1,NBS
 1010 XB(N)=XB(N)-X1
      DO 1020 N=1,NT3
      THETB(N)=THETB(N)/57.2957795
      IF ( N .GT. NT ) THETB(N) = THETB(N - NT + 1 ) + PI
      CPU(N) = COS(THETB(N))
      CPL(N) = SIN(THETB(N))
      COST(N) = COS(THETB(N)+FROLL)
      SINT(N) = SIN(THETB(N)+FROLL)
 1020 CONTINUE
      DO 1040 N=1,NB
      ARC(N) = ARAYAW - DZDX(N)*COS(FROLL)
      RB(N)=(R(N+1)+R(N))/2.
      RB2(N)=RB(N)**2
      ZB(N)=(ZCAM(N+1)+ZCAM(N))/2.
      X(N)=(XB(N+1)+XB(N))/2.
      IF(ARC(N))1030,1040,1030
 1030 KA=1
 1040 CONTINUE
      DO 1060 N=1,NB
      DO 1060 M=1,NB
      EX=X(N)-XB(M)
      EL=XB(NBS)-XB(M)
      D1=SQRT((EX+EL)**2+B2*RB2(N))
      D2=SQRT(EX**2+B2*RB2(N))
      US=1./D1-1./D2
      VS=B2*RB(N)*    (1./(D1*(EX-EL+D1))-1./(D2*(EX+D2)))
 1060 VD(N,M)=VS-DRDX(N)*US
      GO TO 1090
 1070 DO 1080 N=1,NB
      DO 1080 M=1,NB
      EX=X(N)-XB(M)
      EL=XB(NBS)-XB(M)
      D1=((EX-EL)**2+B2*RB2(N))**1.5
      D2=(EX**2+B2*RB2(N))**1.5
      UR=B2*RB(N)*(1./D2-1./D1)
      VR=((EX-EL)*((EX-EL)**2+2.*B2*RB2(N))/D1-EX*(EX**2+2.*B2*RB2(N))
     1/D2)/RB2(N)
 1080 VD(N,M)=VR-DRDX(N)*UR
 1090 CALL SINVRT(51,NB,I1,I2,SC,DET,NDETXP)
      IF(I1)1100,1110,1100
 1100 WRITE (NTAPEO,1590) I1,I2,SC
 1110 IF(KA)1150,1120,1120
 1120 DO 1130 N=1,NB
      Q(N)=0.
      DO 1130 M=1,NB
 1130 Q(M)=Q(M)+VD(N,M)*DRDX(M)
      IF(KA)1170,1170,1140
 1140 KA=-1
      GO TO 1070
 1150 DO 1160 N=1,NB
      QC(N)=0.
      DO 1160 N=1,NB
 1160 QC(N)=QC(N)-VD(N,M)*ARC(M)
 1170 CONTINUE
C
C     VELOCITY COMPONENTS ON THE BODY
C
      DO 1210 N=1,NB
C
      U(N)=0.
      UC(N)=0.
      V(N)=0.
      VC(N)=0.
      W(N)=0.
      DO 1190 M=1,NB
      EX=X(N)-XB(M)
      EL=XB(NBS)-XB(M)
      XL=EX-EL
      D1=SQRT(XL**2+B2*RB2(M))
      D2=SQRT(EX**2+B2*RB2(M))
      D3=D1**3.
      D4=D2**3.
      US=1./D1-1./D2
      VS=B2*RB(N)*(1./(D1*(XL+D1))-1./(D2*(EX+D2)))
      U(N)=U(N)+US*Q(M)
      V(N)=V(N)+VS*Q(M)
      IF(KA)1180,1190,1180
 1180 UR=B2*RB(N)*(1./D4-1./D3)
      VR=(XL*(XL**2+2.*B2*RB2(N))/D3-EX*(EX**2+2.*B2*RB2(N))/D4)/RB2(N)
      WR=(XL/D1-EX/D2)/RB2(N)
      UC(N)=UC(N)+UR*QC(M)
      VC(N)=VC(N)+VR*QC(M)
      W(N)=W(M)+WR*QC(M)
 1190 CONTINUE
      DO 1200 L=1,NNT
      UB(L)=U(N)+UC(N)*COST(L)
      VB(L)=V(N)+VC(N)*COST(L)
 1200 WB(L)=W(N)*SINT(L)
C
      CALL FORMOM(NBS,XB,RB,XMACH,COSA,SINA,COSB,SINB,UB,VB,WB,CPU,
     1          CPL,CPB,N,NNT,CPC)
      IF ( NNT .GT. NT ) CPB(N, NT3) = CPB(N, 1)
C
 1210 CONTINUE
C
      IF ( BROLL .LT. 0.01 ) GO TO 600
C
C         TRANSFORM CPB FROM FLOW COORDINATE TO BODY COORDINATE TO
C        INCORPORATE THE EFFECT OF ROLL (BROLL)
C
      NVAR = 2
      NPTS = NT3
      DO 500 I=1,NB
      DO 510 L=1,NT3
      TX(1,L) = THETB(L)
      TX(2,L) = CPB(I, L)
  510 CONTINUE
      CALL SPLFIT
      DO 520 L=1,NNT
      TY(1) = THETB(L) - BROLL
      IF ( TY(1) .LT. 0.0 ) TY(1) = 2.0*PI + TY(1)
      IF ( TY(1) .GT. 2.0*PI ) TY(1) = TY(1) - 2.0*PI
      CALL POINT(1)
      CPB(I, L) = TY(2)
  520 CONTIKUE
  500 CONTINUE
```

```
C
  600 CONTINUE
C
C     VELOCITY COMPONENTS ON THE WING
      IF(NW)1300,1300,1220
 1220 IF(NSTOR)1230,1240,1230
 1230 NS=1
      JM=2
      SY(1)=1.
      SY(2)=-1.
      GO TO 1250
 1240 NS=NBOD+1
      JM=1
      SY(1)=0.
 1250 CONTINUE
      L=0
C
      DO 1290 N=NS,NP
C
      L=L+1
      U(L)=0.
      V(L)=0.
      W(L)=0.
      EN(L)=0.
      DO 1290 J=1,JM
      DYC=YC(N)-SY(J)*YN
      DZC=ZC(N)-ZN
      IF(DYC.EQ.0..AND.DZC.EQ.0.) GO TO 1260
      THETN=ATAN2(DYC,DZC)
      GO TO 1270
 1260 THETN=0.
 1270 COSTN=COS(THETN)
      SINTN=SIN(THETN)
      R2=(YC(N)-SY(J)*YN)**2+(ZC(N)-ZN)**2
      RR(L)=SQRT(R2)
      BR2=B2*R2
      VR=0.
      VT=0.
      DO 1280 M=1,NB
      EX=XC(N)-XN-XB(M)
      EL=XB(NBS)-XB(M)
      XL=EX-EL
      D1=SQRT(XL**2+BR2)
      D2=SQRT(EX**2+BR2)
      D3=D1**3
      D4=D2**3
      U(L)=U(L)+(1./D1-1./D2)*Q(M)+B2*RR(L)*(1./D4-1./D3)*COSTN*QC(M)
      VR=VR+B2*RR(L)*(1./(D1*(XL+D1))-1./(D2*(EX+D2)))*Q(M)+(XL*(XL**2+2
     1.*BR2)/D3-EX*(EX**2+2.*BR2)/D4)/R2*QC(M)*COSTN
 1280 VT=VT+(XL/D1-EX/D2)/R2*QC(M)*SINTN
      NWS = ( N - NS) / NWPW + 1
      THETS = THETN + TA(N)
      V(L)=VR*SINTN+VT*COSTN+V(L)
      W(L)=VR*COSTN-VT*SINTN+W(L)
      EN(L)=EN(L)+VR*COS(THETS)-VT*SIN(THETS)
C
 1290 CONTINUE
C
 1300 CONTINUE
      IF(EM)1390,1380,1370
 1370 CPSTAG = Q000*((1.+Q001)**3.5-1.)
      GO TO 1390
 1380 CPSTAG=1.
 1390 CONTINUE
      BCL=0.
      BCD=0.
      BCM=0.
      IF(NS1)1410,1410,1400
 1400 IF(NS2)1410,1410,1420
 1410 NS1=1
      NS2=NBS
 1420 NBL=NS2+1-NS1
      NBT=NBL+1
      M1=NS2-1
      MM = 1
      DO 1440 M = NS1,M1
      MM = MM + 1
      RM = RB(M)
      DO 1430  N = 1,NNT
C
      CPBB = CPB(M,N)
      CP(N) = CPBB
      ARC(N) = CPBB * CPL(N)
      YA(N) =  CPBB * CPU(N)
 1430 YX(N) = YA(N) * CPU(N)
      CALL SIMP (GM,THETB,CP,NNT,IRR)
      CALL SIMP (AM,THETB,YA,NNT,IRR)
      CALL SIMP (BM,THETB, YX,NNT,IRR)
      CALL SIMP(ARCM,THETB, ARC, NNT, IRR)
      IF (IRR .NE. 1) GO TO 1580
      A(MM) = AM * RM
      B(MM) = BM * RM
      G(MM) = GM * RM
      ARC(MM) = ARCM * RM
 1440 CONTINUE
      NN = NS1 - 1
      DO 1450 N = 2,NBL
      NN = NN + 1
      C(N) = DZDX(NN) * A(N) + DRDX(NN) * G(N)
 1450 XX(N) = X(NN)
      XX(1) = XB(NS1)
      XX(NBT) = XB(NS2)
      C(1) = .5 * CPSTAG * DS1
      C(NBT) = .5 * CPSTAG * DS2
      IF (R(NBS) .NE. 0.)  C(NBT) = 0.
      A(1) = ZCAM(1)*C(1)
      A(NBT) = ZCAM(NBS)*C(NBT)
      CALL SIMP (CZ,XX,A,NBT,IRR)
      CALL SIMP( CY, XX, ARC, NBT, IR)
      CN=-CN
      IF (IRR .NE. 1) GO TO 1580
      CALL SIMP (CCX,XX,C,NBT,IRR)
      IF (IRR .NE. 1) GO TO 1580
      NN = NS1 - 1
      DO 1460 N = 2,NBL
      NN = NN + 1
      ARC(N) = ARC(N) * ( X(NN) + XN - XP )
 1460 A(N) = A(N)*(X(NN)+XN-XP+RB(NN)*DRDX(NN))+ZB(NN)*C(N)+RB(NN)
     * *DZDX(NN)*B(N)
      CALL SIMP( BBCMY, XX, ARC, NBT, IRR)
      CALL SIMP (CM,XX,A,NBT,IRR)
      IF (IRR .NE. 1) GO TO 1580
      IF ( YAW .LT. 0.01 .AND. BROLL .LT. 0.01 ) GO TO 1465
      GO TO 1466
 1465 CONTINUE
      CCX = 2.0*CCX
      CY = 0.0
      CZ = 2.0*CZ
 1466 CONTINUE
      CCY = -CY
      CCZ = -CZ
      BBCL = -CCX*SINA*COSB + CCY*(SING*COSA+COSG*SINB*SINA) + CCZ*
     1        (COSG*COSA -SING*SINB*SINA)
      BBCD = CCX*COSA*COSB + CCY*(SING*SINA-COSG*SINB*COSA) + CCZ*
     1        (SING*SINB*COSA + COSG*SINA)
      BBCD = CCX*SINB + CCY*COSG*COSB -CCZ*SING*COSB
      BCM = CM + CX*(ZN-ZP)
      DO 1480 N=1,NBS
 1480 XB(N)=XB(N)+X1
C
      NT = NNT
      DO 1490 N=1,NT
 1490 THETB(N)=THETB(N)*57.2957795
C
 1570 CONTINUE
C     PRINT OUT VELOCITY COMPONENTS
C
      WRITE(NTAPEC) NBS
      WRITE(NTAPEC) (XB(J),R(J),J=1,NBS)
      WRITE(NTAPEC) (Q(J),QC(J),J=1,NBS)
C
C ------------------
      RETURN
C ------------------
C.
 1580 WRITE (NTAPEO,1690) IRR
      GO TO 1570
C
C     FORMAT STATEMENTS
C
 1590 FORMAT (1H1,5X,16H ERROR CODE 1 = I5,16H ERROR CODE 2 = I5,
    115H SCALE FACTOR = E16.8)
 1690 FORMAT (1H0,*INTEGRATION ERROR, IRR =*,I10)
C
      END
      SUBROUTINE FORMOM(NBODYS,XB,R,XMACH,COSA,SINA,COSB,SINB,UB,VB,VTB
     1        ,COSTB,SINTB,CPBB,I,NTHETA,CPCALC)
C
C     CALCULATION OF PRESSURES, FORCES, AND MOMENTS
C
      DIMENSION XB(1), R(1), UB(1), VB(1), VTB(1), COSTB(1), SINTB(1),
     1          CPBB(51, 1)
C
      BODYL=XB(NBODYS)-XB(1)
      XMACH2 = XMACH**2
      CPSTAG = 1.42857 * ((1.+.2*XMACH2)**3.5 - 1.)/XMACH2
C
      DO 1420 J=1, NTHETA
      VPM = VTB(J) * COSTB(J)  + VB(J) * SINTB(J)
      WPM = VB(J) *  COSTB(J)  - VTB(J) * SINTB(J)
      UWPM  = UB(J)*COSB*COSA - VPM*SINB*COSA + WPM*SINA
C
      UWIND = 1. + UWPM
      VWIND = UB(J)*SINB + VPM*COSB
      WWIND = -UB(J)*COSB*SINA + VPM*SINB*SINA + WPM*COSA
C
      TWIND2 = UB(J)**2 + VPM**2 + WPM**2 - UWPM**2
C
      IF ((CPCALC - 1.)  1370,1380,1390
 1370 CPBB(I,J) = -2. * UWPM
      GO TO 1400
 1380 XM2 = XMACH2 - 1.
      CPBB(I,J) = -2.*UWPM + XM2*UWPM**2 - TWIND2
      GO TO 1400
 1390 Q2 = UWIND ** 2   + VWIND **2  + WWIND ** 2
```

```
      CPBB(I,J) = 1.42857*((1.+.2*XMACH2*(1.-Q2))**3.5-1.)/XMACH2
 1400 CONTINUE
 1410 CONTINUE
      IF(R(NBODYS).EQ.0.) CPBB(NBODYS,J)=CPSTAG
 1420 CONTINUE
C
      RETURN
      END
      SUBROUTINE CUBERT
      DIMENSION X(3)
      COMMON/LOCTN / V(3), L, M, G(4), VMAX, VMIN
      COMMON/POINTS/ NPTS, NVAR
      L=0
      SUM = SQRT( G(1)**2+G(2)**2+G(3)**2+G(4)**2 )
      G1 = G(1) / SUM
      G2 = G(2) / SUM
      G3 = G(3) / SUM
      G4 = G(4) / SUM
C
C        EQ. 1   G1*X**3 + G2*X**2 + G3*X = 0.
      IF ( ABS( G4) .LT. 1.E-6) GO TO 4
C
C        EQ. 2   G1*X**3 + G2*X**2 + G3*X +G4 = 0.
      IF ( ABS( G1) .GT. 1.E-6) GO TO 9
C
C        EQ. 3   G2*X**2 + G3*X + G4 = 0.
      IF ( ABS( G2) .GT. 1.E-6) GO TO 7
C
C        EQ. 4   G3*X + G4 = 0.
      IF ( ABS( G3) .GT. 1.E-6) GO TO 5
      RETURN
C
  4   CONTINUE
      V(1) = 0.0
      L=1
      RETURN
  5   CONTINUE
      V0   = - G(4) / G(3)
      GO TO 35
  7   CONTINUE
      R1 = G(3)**2 - 4.0*G(2)*G(4)
      IF (  R1  .GT. 0.0 )  GO TO 107
      WRITE (6,901)
  901 FORMAT (1H0,*NAGATIVE RADICAL RESULTED FROM SOLVING THE QUADRATIC
     1EQUATION  G2.X2 + G3.X + G4 = 0*)
      RETURN
  107 CONTINUE
      R1=SQRT(R1)
      V1 = ( -G(3) + R1 ) *0.5 / G(2)
      V(1) = V1
      IF ( V1  .GT. VMIN .AND.  V1  .LT. VMAX ) GO TO 8
      V2 = (-G(3) - R1 )*0.5 / G(2)
      V(2) = V2
      IF ( V2  .GT. VMIN .AND.  V2  .LT. VMAX ) GO TO 8
      WRITE (6,902) VMIN, V1,V2, VMAX
  902 FORMAT (1H0,*BOTH ROOTS V1, V2 OF QUADRATIC EQ. G2.X2 + G3.X + G4
     1= 0 OUTSIDE OF LIMIT VMIN, VMAX*/ 1X,*VMIN =*,E14.4,4X,*V1 =*,
     2          E14.4,4X,*V2 =*,E14.4,4X,*VMAX =*,E14.4)
      RETURN
  8   CONTINUE
      L = 1
      RETURN
C
C        THREE ROOTS
C
```

```
    9 CONTINUE
      A1 = G(2) / G(1)
      A2=G(3)/G(1)
      A3=G(4)/G(1)
      A=A2-A1*A1/3.
      Q=2.*A1*A1*A1/27.-A1*A2/3.+A3
      RAD=Q*Q/4.+A*A*A/27.
      SIGN=1.
      IF(Q.GE.0.)SIGN=-1.
      IF(RAD)10,20,20
C
C                THREE UNEQUAL REAL ROOTS
C
   10 CONTINUE
      A3 = SQRT( ABS( A/3.0) )
      A2 = SQRT( ABS( A**3/3. ) )
      PHI = ACOS( ABS( 1.5*Q/ A2) )
      DO 11 I=1,3
      X(I) = 2.0*SIGN*A3*COS((PHI+6.2831853*FLOAT(I-1))/3.0)
   11 CONTINUE
      GO TO 40
C
C        EITHER ONE REAL AND TWO IMAG. OR THREE REAL ROOTS OF WHICH TWO
C        ARE EQUAL
C
   20 CONTINUE
      V1 = 1.0 / 3.0
      Q = A2*V1 - ( A1*V1)**2
      R = .5*A1*A2*V1 - .5 *A3 - ( A1*V1) **3
      RAD = Q**3 + R*R
      IF ( RAD  .GT.  0.0 )  GO TO 120
      WRITE (6,903)
  903 FORMAT (1H0,*ERROR IN TESTING RADICAL, REDUNDENT TEST*)
      RETURN
  120 CONTINUE
      RAD = SQRT( RAD)
      A2 = R + RAD
      SIGN = 1.
      IF ( A2  .LT.  0.0)  SIGN = -1.
      A2 = SIGN * ABS( A2 ) **V1
      A3 = R - RAD
      SIGN = 1.
      IF ( A3 .LT. 0.0 ) SIGN = -1.
      A3 = SIGN * ABS( A3 )**V1
      VO = A2 + A3 - A1*V1
   35 CONTINUE
      IF ( VO .GT. VMIN .AND. VO  .LT. VMAX ) L = 1
      IF ( L  .EQ. 0 ) WRITE (6,905)
  905 FORMAT (1H0,*THE ONLY UNEQUAL ROOT IS OUTSIDE OF VMIN, VMAX*/1X,
     1            *VMIN =*,E14.4,4X,* V =*,E14.4,4X,*VMAX =*,E14.4)
      V(1) = VO
      RETURN
   40 CONTINUE
      DO 45 I=1,3
      VO = X(I) - A1/3.0
      X(I) = VO
      V(I) = VO
      IF ( VO .LT. VMIN .OR.  VO  .GT. VMAX ) GO TO 50
      L=1
      RETURN
   50 CONTINUE
   45 CONTINUE
      WRITE (6,904) VMIN, V(1), V(2), V(3), VMAX
  904 FORMAT (1H0,*ALL THREE UNEQUAL ROOTS OUTSIDE OF LIMIT VMIN, VMAX*/
     1     1X,*VMIN =*,E14.4,4X,*V1 =*,E14.4,4X,*V2 =*,E14.4,4X,*V3 =*,
     2          E14.4,4X,*VMAX =*,E14.4)
C
      RETURN
      END
C
      SUBROUTINE POINT( N )
      COMMON/CURFIT/ B(2,4, 50), T( 50), SCALEF(2)
      COMMON/ ENDPT/ XFIRST, XLAST, XSAVE( 50)
      COMMON/LOCTN/ V(3), L, M, A(4), VMAX, VMIN
      COMMON/POINTS/ NPTS, NVAR
      COMMON/ POINTP/  X(4),  XP( 4),  XPP( 4)
      DIMENSION DX(4)
C
      N1 = NPTS - 1
      IF(N.NE.0)GO TO 4
      DO 1 I=1,N1
      M=I
      IF ( V(1)  .LE. T(I+1) ) GO TO 2
    1 CONTINUE
      WRITE(6,100)
  100 FORMAT(1H0,33H ERROR - PARAMETER EXCEEDS LIMITS)
      RETURN
    2 CONTINUE
      DO 300 K=1,3
      P = V(K)
      DX(K) = 1.E10
      IF ( P .LT. VMIN  .OR.  P  .GT. VMAX ) GO TO 300
      DX(K) = ABS((((B(1,1,M)*P + B(1,2,M))*P + B(1,3,M))*P + B(1,4,M))*
     1          SCALEF(1) - XIND )
  300 CONTINUE
      IF ( DX(1)  .GT. DX(2) ) GO TO 302
      IDENT = 1
      IF ( DX(1)  .GT. DX(3) )  IDENT = 3
      GO TO 305
  302 CONTINUE
      IDENT = 2
      IF ( DX(2)  .GT. DX(3) )  IDENT = 3
  305 CONTINUE
      P = V(IDENT)
C
      DO 3   I=1, NVAR
      X(I)=(((B(I,1,M)*P+B(I,2,M))*P+B(I,3,M))*P+B(I,4,M))*SCALEF(I)
      XP(I)=((3.*B(I,1,M)*P+2.*B(I,2,M))*P+B(I,3,M))*SCALEF(I)
    3 XPP(I)=(6.*B(I,1,M)*P+2.*B(I,2,M))*SCALEF(I)
C
  310 CONTINUE
      RETURN
C
    4 X(N)=X(N)/SCALEF(N)
      DO 5 I=1,N1
      M=I
      K = I + 1
      IF ( XSAVE(K)  .GE. X(N) ) GO TO 6
    5 CONTINUE
      XIND = X(N) * SCALEF(N)
      XFT = XFIRST * SCALEF(1)
      XLT = XLAST * SCALEF(1)
      WRITE (6,101)  XIND, XFT, XLT
  101 FORMAT (1H0,*  THE GIVEN VALUE OF INDEPENDENT VARIABLE *,E12.4,
     1          5X,*IS OUT OF RANGE TO BE INTERPOLATED  *,E12.4,5X,E12.4)
      RETURN
    6 DO 7 I=1,4
    7 A(I)=B(N,I,M)
      A(4)=A(4)-X(N)
      IF ( XFIRST  .EQ. X(N) ) GO TO 112
      IF ( XLAST   .EQ. X(1) ) GO TO 111
      VMAX = T(M+1) * 1.0001
      VMIN = T(M) * 0.9999
      CALL CUBERT
      XIND = X(1)* SCALEF(1)
      GO TO 103
  112 V(1) = 0.0
      L=1
      GO TO 103
  111 V(1) = 1.0
      L=1
  103 CONTINUE
      IF(L.NE.0)GO TO 2
      OP = P
      WRITE (6,10) OP
   10 FORMAT (1X4HOP =,E15.8)
      DO 8 J=1,2
      K = M + J - 1
      B1 = B(N,1,K)
      B2 = B(N,2,K)
      B3 = B(N,3,K)
      B4 = B(N,4,K)
      P = T(K)
      SUM = B1*P**3 + B2*P**2 + B3*P + B4
      DIF = SUM - X(N)
    8 WRITE (6,9) K,P,B1,B2,B3,B4,X(N),SUM,DIF
    9 FORMAT (1XI3,8E15.8)
      WRITE(6,102)
  102 FORMAT(1H0,32H FAILURE IN CUBE ROOT EXTRACTION)
      RETURN
      END
C
      SUBROUTINE SPLFIT
      REAL L,M,MU
      COMMON/CURFIT/ B(2,4, 50), T( 50), SCALEF(2)
      COMMON/ ENDPT/ XFIRST, XLAST, XSAVE( 50)
      COMMON/INPUT /  X( 2,  50)
      COMMON/POINTS/ NPTS, NVAR
      DIMENSION H(2, 50), S( 50), L( 50), MU(.50), D( 50), P( 50), Q( 50
     1          ), U( 50)
      IF ( NPTS .GT.  50) NPTS= 50
      DO 200 I=1, NVAR
  200 SCALEF(I)=0.
      DO 201 I=1,NPTS
      DO 201 J=1,NVAR
  201 SCALEF(J)=AMAX1(SCALEF(J),ABS(X(J,I)))
      DO 101 J=1,NVAR
      IF ( SCALEF(J)  .LT. 1.E-10 ) SCA'EF(J) = 1.0
  101 CONTINUE
      DO 202 I=1,NPTS
      DO 202 J=1,NVAR
  202 X(J,I)=X(J,I)/SCALEF(J)
      DO 203 I=1,NPTS
      XSAVE(I) = X(1,I)
  203  CONTINUE
      XFIRST = X(1,1)
      XLAST = X(1,NPTS)
      T(1)=0.
      S(1)=0.
      SUM=0.
      L(1)=-2.
      MU(1)=0.
      L(NPTS)=0.
      MU(NPTS)=-2.
      D(1)=0.
      D(NPTS)=0.
      P(1)=2.
      Q(1)=1.
      U(1)=0.
      N1=NPTS-1
      DO 2 I=2,NPTS
      SUM1=0.
      DO 1 J=1,NVAR
    1 SUM1=SUM1+(X(J,I)-X(J,I-1))**2
      S(I)=SQRT(SUM1)
    2 T(I)=T(I-1)+S(I)
      DO 3 I=2,NPTS
      S(I)=S(I)/T(NPTS)
    3 T(I)=T(I)/T(NPTS)
      DO 4 I=2,N1
      L(I)=S(I+1)/(S(I)+S(I+1))
      MU(I)=1.-L(I)
      P(I)=MU(I)*Q(I-1)+2.
    4 Q(I)=-L(I)/P(I)
      P(NPTS)=2.*(1.-Q(N1))
      DO 7 I=1,NVAR
      DO 5 J=2,N1
      D(J)=6.*((X(I,J+1)-X(I,J))/S(J+1)-(X(I,J)-X(I,J-1))/S(J))/(S(J)+
     1S(J+1))
    5 U(J)=(D(J)-MU(J)*U(J-1))/P(J)
      U(NPTS)=-MU(NPTS)*U(N1)/P(NPTS)
      M(I,NPTS)=U(NPTS)
      DO 6 J=1,N1
      K=NPTS-J
      M(I,K)=Q(K)*M(I,K +1)+U(K)
      B(I,1,K)=(M(I,K +1)-M(I,K))/(6.*S(K+1))
      B(I,2,K)=.5*(M(I,K)*T(K+1)-M(I,K+1)*T(K))/S(K+1)
      B(I,3,K)=(.5*(M(I,K+1)*T(K)**2-M(I,K)*T(K+1)**2)+X(I,K+1)-X(I,K)+
     1(M(I,K)-M(I,K+1))*S(K+1)**2/6.)/S(K+1)
    6 B(I,4,K)=(M(I,K)*T(K+1)**3-M(I,K+1)*T(K)**3)/(6.*S(K+1))+(X(I,K)*T
     1(K+1)-X(I,K+1)*T(K))/S(K+1)+S(K+1)/6.*(M(I,K+1)*T(K)-M(I,K)*T(K+1)
     1)
    7 CONTINUE
      RETURN
      END
C
      OVERLAY(WANG, 7, 0)
      PROGRAM WANG76
C
      COMMON/ MAIN / NTAPEA,NTAPEB,NTAPEC,NTAPED,NTAPEE,NTAPEF,NTAPEI,
     1          NTAPEO,NBODY,NWING,XMACH,STM,KACE,NPOLAR,IRW
      COMMON / BODYSP/ DRDX(51), DZDX(51), DRDXC(51)
      COMMON /AVAR/  A(210),ACB(21),ABX(100),AWX(110),AREA(210),ARW(2),A
     1               RWT(20),ALPHA(210),ALPHAB(210),ALPHAT(110),
     2               ALPHAW(110),ALPHAX(110),AWS(110),ABT(100),ANIN(
     3               210,2),ALPHAA,ALPHAD,ARA,ARADEG,ARB,ARW(10), ARAS,
     4               AT, AAA
      COMMON /BODS/ NUMBOD,XNOS(3),YNOS(3),ZNOS(3),NBSTAT(3)
      COMMON /BVAR/  B(210), BBCL, BBCD, BBCY, BBCM, BBCMR, BBCMY
      COMMON/CVAR/C(210), CHORD(210),CL(210),CPNN(1,1,1)
     1               ,CLS(210)
     2               ,CDR,CASE,CPCALC,CBAR,CONSNT,CLBAR,CLX,CLM,CON
      COMMON /KVAR/  KASE,KONFIG,KPOLAR
      COMMON /NVAR/  NFMT(9),NROW(2)
     1         ,NS,NPANEL,NACEL,NROWB,NROWW,NCOLW,NTHETA,NTHETS,
     2         NXLE, NRG, NPOLR, NCLX
      COMMON/RVAR/ R(51), RN(51,2), RATIOX, RFAREA
      COMMON/TVAR/ TITLE(20), THETA(210), THETAB(15),THETAS(15), THETAN
     1         (11,2), TCL(11), TCD(11), TCM(11), THICK,TWIST
      COMMON /UVAR/  UBWT(150),UWBT(110),UWWT(110),UNCL(1,1)
      COMMON /VVAR/  VBWT(150),VWBT(110),VWWT(110),VNCL(1,1)
     1               ,VOUT
      COMMON /WVAR/  WBWT(150),WWBT(110),WWWT(110),WNCL(1,1)
      COMMON/XVAR/ XBAR(210), XC(210), XNI(2), XNXN(2), XNTN(2), XB(51),
     1         XN(51,2), XNNCD(2), XNNCL(2), XNNCM(2), XYZ(3), XCL(11),
     2         XCD(11), XNACEL, XP, XCPBAR
      COMMON /YVAR/  YBAR(210),YC(210),YNI(2)
      COMMON/ZVAR/ ZBAR(210), ZC(210), ZNI(2), ZDELTA(51), ZDN(51,2)
     1               ,ZP,ZA
      COMMON/PBDATA/ NBODYS, XMIDB(51), NTHETB, THETB(15), CPBB(51,15)
     1         ,XCB(3,700),NG(350),NELM(700),NE,NGRIDP,PINF,IPUNCH
     2         , CPB
      COMMON/EXTRA/ ARAN, DSB1, DSB2, DS1, DS2, NB1, NB2, NS1, NS2,
     1         NTN, NXN, XNOSE
      COMMON/NSLOPE/ DRNDX(51,2), DZNDX(51,2)
      COMMON/WINGOT/ CPU, CPL,NWG,ISOLID,IFORM(10),YAW,BROLL
      DIMENSION NS1(2), NS2(2), DS1(2), DS2(2)
      DIMENSION CPB(150), CPU(110), CPL(110)
C
C
      IF ( KONFIG  .EQ. 1 ) GO TO 50
      CALL KARMOR(NBODYS, NTHETB, 0, XNOSE, 0.0, 0.0, CPCALC, VOUT,ARA,
     1         XP, ZP, RFAREA, XB, R, THETB, ZDELTA, XC, YC, ZC, THETA,
     2         UWBT, VWBT, WWBT, ALPHAX, CPBB, NB1, NB2, DSB1, DSB2)
C
C
   50 CONTINUE
      IF ( XNACEL ) 100, 500, 100
  100 CONTINUE
C
      DO 200 N=1, NACEL
C
      DO 150 J=1,NXN
      DRDX(J) = DRNDX(J,N)
      DZDX(J) = DZNDX(J,N)
  150 CONTINUE
C
      ARAN = ARA + ARN(N) / COR
      CALL KARMOR( NXN, NTN, NACEL, XNI(N), YNI(N), ZNI(N), CPCALC, VOUT
     1         , ARAN, XP, ZP, RFAREA, XN(1,N), RN(1,N), THETAN(1,N),
     2         ZDN(1,N), XC, YC, ZC, THETA, UNCL(1,N), VNCL(1,N),
     3         WNCL(1,N), AN1N(1,N), CPNN(1,1,N), NS1(N), NS2(N),
     4         DS1(N), DS2(N) )
  200 CONTINUE
C
  500 CONTINUE
C
      END
C
      SUBROUTINE KARMOR(NBODYS,NTHETA,NACEL,XN,YN,ZN,CPCALC,VOUT,ARA
     1,XP,ZP,RFAREA,XB,R,THETAB,ZDELTA,XC,YC,ZC,THETA,U,VV,WW,AN1,CPBB
     2         ,NS1,NS2,DS1,DS2)
C
C ......................................................................
C     CALCULATION OF VELOCITY COMPONENTS DUE TO BODY THICKNESS
C ......................................................................
C
C
      COMMON/ MAIN / NTAPEA,NTAPEB,NTAPEC,NTAPED,NTAPEE,NTAPEF,NTAPEI,
     1         NTAPEO,NBODY,NWING,IMACH,SYM,KACE,NPOLAR,IRW
      COMMON / BODYSP/ DRDX(51), DZDX(51), DRDXC(51)
      COMMON/WINGOT/ CPU, CPL,NWG,ISOLID,IFORM(10),YAW,BROLL
      COMMON /DVAR/  D(210), DBCL, DBCD, DBCD, DBCM, DBCMR, DBCMY
      COMMON/INPUT / TXX(2,  50)
      COMMON/ POINTP/ TY(4), TYP(4), TYPP(4)
      COMMON/ POINTS/ NPTS, NVAR
C
      DIMENSION XB(1),R(1),THETAB(1),ZDELTA(1),XC(1),YC(1),ZC(1)
     1          ,U(1), VV(1), WW(1), AN1(1), CPBB(51,1), THETA(1)
      DIMENSION  ALPHAC(51), T(51),TB(15), TC(51), TX(51), UB(22),VB(22)
     1          , VTB(22), SY(2), FB(51), SD(51), NBS(11), COSTB(22),
     2          SINTB(22), CP(150), YA(22), YB(22), A(51), B(51), G(51),
     3          C(51), XBB(51), CPU(110), CPL(110)
C
C
      EPS = 1.0E-6
      PI=3.1415926
      XMACH2=XMACH*XMACH
      BT2=XMACH2-1.
      BETA=SQRT(BT2)
      NWPW = NWING / NWG
      NYAW = 0
      NT = NTHETA
      NT1 = NT + 1
      NT2 = 2*( NT - 1 )
      N1=NBODYS-1
      X1 = XB(1)
      NT3 = NT2 + 1
      TANA = TAN(ARA)
      COSAR = COS(ARA)
      SINAR = SIN(ARA)
      TANY = TAN(YAW)
      COSY = COS(YAW)
      SINY = SIN(YAW)
      A1 = 1.0/SQRT( 1.0 + TANA**2 + TANY**2 )
      COSB = A1*SQRT( 1.0 + TANA**2 )
      SINB = SQRT( 1.0 - COSB**2 )
      COSG = COS(BROLL)
      SING = SIN(BROLL)
C
      DO 1000 N = 1, NBODYS
 1000 XB(N) = XB(N) - X1
C
C        INCORPORATE EFFECT OF YAW TO ANGLE OF ATTACK
C
      ARAYAW = ACOS( A1 )
      SINAY = SIN(ARAYAW)
      ARJ = ARA
      IF ( YAW .LT. 0.01 .AND. BROLL .LT. 0.01 ) GO TO 100
      NT = NT2
      NT3 = NT + 1
      IF ( ARA .LT. 0.01 .AND. YAW .GT. 0.01) GO TO 90
      FROLL = ATAN( TANY / TANA )
      GO TO 150
   90 CONTINUE
      FROLL = 0.5 * PI
      GO TO 150
  100 CONTINUE
      FROLL = 0.0
  150 CONTINUE
      DO 1010 J = 1,NT3
      THETAB(J) = THETAB(J) / 57.2957795
      IF ( J .GT. NTHETA ) THETAB(J) = THETAB(J-NTHETA+1) + PI
C
C        CPU, CPL BASED ON THETAB BEFORE INCORPORATION OF YAW EFFECT
C
      CPU(J) = COS(THETAB(J))
      CPL(J) = SIN(THETAB(J))
      COSTB(J) = COS(THETAB(J)+FROLL)
 1010 SINTB(J) = SIN(THETAB(J)+FROLL)
```

```
C
C     COMPUTE FIRST AND SECOND DERIVATIVES
      DO 1015 N = 1, NBODYS
 1015 ALPHAC(N) = ARAYAW - DZDX(N)*COS(FROLL)
      DO 115 I=1,N1
      FD(I) = DRDX(I)
  115 CONTINUE
      FD2 = FD(1) * FD(1)
      BTAN = BETA * FD(1)
      IF (BTAN .LT. 1.0) GO TO 1020
      WRITE (6,1580)
C
C ******************
      STOP
C ******************
C
 1020 SBT2 = SQRT (1.-BT2*FD2)
      SLOG = ALOG((1.+SBT2)/BTAN)*FD2
C
C     CALCULATE SOURCE STRENGTHS
C
      DO 1150 I=1,N1
C
      M=I+1
      UBSUM=0.
      VBSUM=0.
      UBDSUM=0.
      VBDSUM=0.
      VTDSUM=0.
      PHIS = 0.
      PHID=0.
      RDRDX = FD(M)*R(M)
      RM=R(M)
      RM2=RM*RM
      RDRDM=RDRDX
      RS=RM2*ALPHAC(I)
      TRM1 = XB(M)-XB(1)
      BRM=BETA*RM
      BRM2 = BRM*BRM
      RABS1 = SQRT(TRM1*TRM1-BRM2)
      IF(BRM.EQ.0.)GO TO 1030
      UBS1 = -ALOG ((TRM1 + RABS1)/BRM)
 1030 RRABD1 = -.5*(TRM1*RABS1-BRM2*UBS1)
      RUBD1 = RABS1
      RVTBD1 = -.5*(TRM1*RABS1 + BRM2*UBS1)
      RN1 = RABS1-RDRDX*UBS1
      RM1 = RRABD1-RDRDX*RUBD1
      IF(I.NE.1)GO TO 1050
C
C     IF I EQUALS 1, DO TANGENT CONE SOLUTION
      T11 = FD2/(SBT2 + SLOG)
      TC11 = FD2*ALPHAC(1)/((FD2*.5)*SBT2 +.5*BT2*SLOG)
      DO 1040 K=1,NT
      UB(K) = - SLOG*T11 /FD2
      VB(K) = SBT2*T11/FD(1)
      VTB(K) = 0.
      IF(RM.EQ.0.) GO TO 1040
      UB(K)=UB(K)+COSTB(K)*SBT2/FD(1)*TC11
      VB(K)=VB(K)-.5*COSTB(K)*(SBT2+BT2*SLOG)/FD2*TC11
      VTB(K)=-.5*SINTB(K)*(SBT2-BT2*SLOG)/FD2*TC11
 1040 CONTINUE
C
      CALL FORMOM(NBODYS,XB,R,XMACH,COSAR,SINAR,COSB,SINB,UB,VB,VTB,
     1          CPU,CPL,CPBB,I,NT,CPCALC)
C
```

```
      CPBB(1, NT3) = CPBB(1, 1)
C
 1050 CONTINUE
      IF (RM .EQ. 0) GO TO 1060
       Q001 = RUBD1*TC11/RM
       Q002 = RABS1*T11/RM
       Q003 = RRABD1*TC11/RM2
       Q004 = RVTBD1*TC11/RM2
 1060 DO 1130 J=1,I
      TRMJ = XB(M)-XB(J) + BETA*R(J)
      TRMJ2 = TRMJ*TRMJ
      IF(BRM.EQ.0.) GO TO 1100
      F1=SQRT(TRMJ2-BRM2)
      F2 = ALOG((TRMJ+F1)/BRM)
      RABS2 = TRMJ*F1-BRM2*F2
      UBS2 = 2.*(F1-TRMJ*F2)
      RM2=RABS2-RDRDM*UBS2
      RRABD2 = -((TRMJ2-4.*BRM2)*F1/3.*BRM2*TRMJ*F2)
      RUBD2 = RABS2
      RVTBD2 = -((TRMJ2 + 2.*BRM2)*F1/3.-BRM2*TRMJ*F2)
      RRM2=RRABD2-RDRDM*RUBD2
C
      IF (J .NE. I) GO TO 1080
C
      T(J) = (RDRDX-RM1*T11)/RM2
      TC(J) = -(RS+RM1*TC11)/RRM2
 1080 RDRDX = RDRDX-RM2*T(J)
      RS=RS+RRM2*TC(J)
      IF(J.NE.1) GO TO 1090
      UBSUM = UBS1*T11
      UBDSUM = Q001
      VESUM =  Q002
      VBDSUM = Q003
      VTDSUM = Q004
 1090 CONTINUE
      UESUM = UBSUM+UBS2*T(J)
      VBSUM = VBSUM+RABS2*T(J)/RM
      UBDSUM = UBDSUM + RUBD2*TC(J)/RM
      VBDSUM = VBDSUM+RRABD2*TC(J)/RM2
      VTDSUM = VTDSUM+RVTBD2*TC(J)/RM2
      GO TO 1130
 1100 IF(I.EQ.J) GO TO 1120
      IF (J.NE.1) GO TO 1110
      PHIS = TRM1*T11
      PHID = 1.5*TRM1*TRM1*TC11
 1110 PHIS = PHIS+TRMJ2*T(J)
      PHID = PHID+TRMJ*TRMJ2*TC(J)
      GO TO 1130
 1120 T(J) = -PHIS/TRMJ2
      TC(J) =-PHID/(TRMJ*TRMJ2)
C
 1130 CONTINUE
C
      DO 1140 K=1,NT
      UB(K) = UBSUM+UBDSUM*COSTB(K)
      VB(K) = VBSUM+VBDSUM*COSTB(K)
 1140 VTB(K) = VTDSUM*SINTB(K)
C
      CALL FORMOM(NBODYS,XB,R,XMACH,COSAR,SINAR,COSB,SINB,UB,VB,VTB,
     1            CPU,CPL,CPBB,M,NT,CPCALC)
      IF ( NT .GT. NTHETA ) CPBB(M,NT3) = CPBB(M, 1)
C
 1150 CONTINUE
C
      IF ( BROLL .LT. 0.01) GO TO 600
```

```
C
C          TRANSFORM CPB FROM FLOW COORDINATE TO BODY COORDINATE TO
C         INCORPORATE THE EFFECT OF ROLL (BROLL)
C
      NVAR = 2
      NPTS = NT3
      DO 500 I=1,NBODYS
      DO 510 L=1,NT3
      TXX(1,L) = THETAB(L)
      TXX(2,L) = CPBB(I, L)
  510 CONTINUE
      CALL SPLFIT
      DO 520 L=1,NT
      TY(1) = THETAB(L) - BROLL
      IF ( TY(1) .LT. 0.0 ) TY(1) = 2.0*PI + TY(1)
      IF ( TY(1) .GT. 2.0*PI ) TY(1) = TY(1) - 2.0*PI
      CALL POINT(1)
      CPBB(I, L) = TY(2)
  520 CONTINUE
  500 CONTINUE
C
  600 CONTINUE
C
C     CALCULATION OF VELOCITY COMPONENTS IN THE FIELD (EFFECT OF WING
C        ON BODY)
C
C
      IF (NWING)1160,1360,1160
 1160 CONTINUE
      IF (NACEL.NE.0) GO TO 1170
      NS=NBODY+1
      JM=1
      SY(1)=0.
      GO TO 1180
 1170 NS=1
      JM=2
      SY(1)=1.
      SY(2)=-1.
 1180 CONTINUE
      NPANEL=NBODY+NWING
      II=0
      DO 1350 I=NS,NPANEL
      II=II+1
      U(II)=0.
      VV(II)=0.
      WW(II)=0.
      AN1(II)=0.
      DO 1350 N=1,JM
      XCN=XC(I)-XN
      YCN=YC(I)-SY(N)*YN
      ZCN=ZC(I)-ZN
      IF (YCN.EQ.0.) GO TO 1190
      THETAA = ATAN2( YCN, ZCN )
      GO TO 1220
 1190 IF(ZCN)1210,1200,1200
 1200 THETAA=0.
      GO TO 1220
 1210 THETAA=PI
 1220 CONTINUE
      COSTHA = COS( THETAA )
      SINTHA = SIN( THETAA )
      R2=YCN*YCN+ZCN*ZCN
      IF(R2.LE.EPS) R2 = 0.
      R1=SQRT(R2)
      BR1=BETA*R1
      BR2=BT2*R2
      TRM=XCN-XB(NBODYS)+BETA*R(NBODYS)
      IF(R1.EQ.0.) GO TO 1240
      IF(TRM-BR1)1240,1240,1230
 1230 G1=TRM*TRM-BR2
      TRMO=TRM/BR1
      G2=ALOG(TRMO+SQRT(TRMO*TRMO-1.))
      G3=SQRT(G1)
 1240 CONTINUE
      US=0.
      VS=0.
      UD=0.
      VD=0.
      VT=0.
      VTF = 0.
      VF = 0.
      IF (TRM.LE.BR1) GO TO 1250
       Q001 = 2.*G1*G3
       Q002 = R1*G3
       Q003 = G2*BR2
       Q004 = BR2/G1
       Q005 = 2.*R2
       Q006 = TC11*SINTHA
       Q007 = TRM*G3-BR2*G2
       Q008 = 2.*G3
       Q009 = 3.*G1*G3
       Q010 = 3.*G3
       Q011 = BT2*COSTHA*TC11
       Q012 = BT2*SINTHA*TC11
       Q013 = 2.*TRM
       Q014 = TC11*COSTHA
       Q017 = 2.*BR2
       Q018 = BT2*COSTHA
       Q019 = BT2*SINTHA
 1250 DO 1340 J=1,N1
      BR=BETA*R(J)
      TRM1=XCN-XB(J)+BR
      TRM2=TRM1*TRM1
      IF(BR1.EQ.0.) GO TO 1270
      TRM3=TRM1/BR1
      IF(TRM3.LT.1.) GO TO 1340
C
C     VELOCITY DUE TO QUADRATIC SINGULARITIES
      F1=SQRT(TRM3*TRM3-1.)
      F2=ALOG(TRM3+F1)
      F1=BR1*F1
      US=2.*(F1-TRM1*F2)*T(J)
      VS=(TRM1*F1-BR2*F2)/R1
      UD=VS*TC(J)
      VS=VS*T(J)
      VD=-((TRM2-6.*BR2)*F1/3.+TRM1*BR2*F2)*TC(J)/R2
      VT=-((TRM2+2.*BR2)*F1/3.-TRM1*BR2*F2)*TC(J)/R2
C
C     VELOCITY DUE TO TANGENT CONE
      IF (J.NE.1) GO TO 1260
      US=US-F2*T11
      VS=VS+F1*T11/R1
      UD=UD+F1*TC11/R1
      VD=VD-.5*(TRM1*F1+BR2*F2)*TC11/R2
      VT=VT-.5*(TRM1*F1-BR2*F2)*TC11/R2
 1260 CONTINUE
C
C     SUMMATION OF VELOCITY COMPONENTS
      U(II) = U(II) +US +COSTHA*UD
      VF = VF +VS +COSTHA*VD
      VTF = VTF +VT*SINTHA
 1270 IF(TRM-BR1)1330,1330,1280
 1280 XI = XB(NBODYS)-(XB(J)-BETA*R(J))-BETA*R(NBODYS)
      TRMX=TRM+XI
      IF(R1)1330,1310,1290
 1290 IF(J.NE.1) GO TO 1300
      U(II)=U(II)+T11*(G2+XI/G3)-Q014*(G3+XI*
     1      (1.-XI*BR2/Q001))/R1
      VF = VF - T11*(TRM*TRMX - BR2)/Q002 + Q014*( Q003
     1      +TRMX*G3-XI*(G3-TRM*(2.+XI*(1.-Q004  )/G3))))/Q005
      VTF=VTF+Q006 *(Q007 +XI*TRM*(2.+XI/G3))/Q005
 1300 CONTINUE
      U(II)=U(II)+2.*T(J)*(TRMX*G2-G3+XI*XI/Q008    )
     1   -TC(J)*((TRMX+XI)*G3-Q003  +XI*XI*(1.-XI*BR2/Q009 ))/R1
     2    *COSTHA
      VF=VF-T(J)*((TRMX+XI)*G3-Q003  +TRM*XI*XI/G3)/R1
     1+TC(J)*COSTHA*((TRMX*TRMX-4.*BR2)*G3/3.+TRMX*Q003  +XI*(TRM-XI)
     2*G3/3.+XI*XI*TRM*(1.+XI*(1.-Q004  )/Q010   ))/R2
      VTF=VTF+TC(J)*SINTHA*((TRM2*Q017  )*G3/3.0
     1-TRMX*Q003  +XI*(TRM-XI)*G3/3.+XI*XI*TRM*(1.+XI/Q010   ))/R2
      GO TO 1330
 1310 IF(TRM.EQ.0.) GO TO 1330
      G4 = ALOG(TRMX/TRM)
      IF(J.NE.1) GO TO 1320
      U(II)=U(II)+T11*(XI/TRM-G4)
      G5=-(G4+XI*XI/(2.*TRM2))/2.
      VF=VF+Q011 *G5
      VTF=VTF-Q012 *G5
 1320 U(II)=U(II)+2.*T(J)*(XI-TRMX*G4+XI*XI/Q013    )
      G6=TRMX*G4-XI*(1.-XI*XI/(6.*TRM2))
      VF=VF-Q018*TC(J)*G6
      VTF=VTF+Q019*TC(J)*G6
 1330 CONTINUE
 1340 CONTINUE
C
      VV(II) = VF*SINTHA+VTF*COSTHA + VV(II)
      WW(II) = VF*COSTHA-VTF*SINTHA+WW(II)
      THETAS = THETAA + THETA(I)
      AN1(II) = VF*COS(THETAS)-VTF*SIN(THETAS)+AN1(II)
C
 1350 CONTINUE
 1360 CONTINUE
C
      BBCL=0.
      BBCD=0.
      BBCM=0.
      CPSTAG = 1.42857*((1.0+0.2*XMACH2)**3.5 - 1.0) / XMACH2
      IF (NS1 .EQ. 0) NS1 = 1
      IF (NS2 .EQ. 0) NS2 = NBODYS
      NBT = NS2 - NS1 + 1
      DTHETB = PI/(NTHETA-1)
      MM = 0
      DO 1460 M = NS1, NS2
      MM = MM + 1
      XBB(MM)=XB(M)
      RM=R(M)
      DO 1630 N=1,NT3
      CPB = CPBB( M, N)
      CP(N) =CPB
      TC(N) = CPB*CPL(N)
      YA(N)=CPB*CPU(N)
 1630 YB(N)=YA(N)+CPU(N)
      CALL SIMP(GM,THETAB,CP,NT3,IRR)
      CALL SIMP(DM,THETAB,YB,NT3,IRR)
      CALL SIMP(AM,THETAB,YA,NT3,IRR)
      CALL SIMP(TCM,THETAB,TC,NT3,IR)
      IF(IRR.NE.1) GO TO 1570
      A(MM) = AM*RM
      D(MM) = DM*RM
      G(MM) = GM*RM
      SD(MM) = TCM*RM
 1440 CONTINUE
      MM = 0
      DO 1450  N = NS1, NS2
      MM = MM + 1
      C(MM) = DZDX(N)*A(MM) + FD(N)*G(MM)
      IF (N .EQ. 1)  C(MM) = .5*CPSTAG*DS1
      IF (N .EQ. NBODYS .AND. R(NBODYS) .EQ. 0.)  C(MM) = .5*CPSTAG*DS2
 1450 CONTINUE
      CALL SIMP(CX,XBB,C,NBT,IRR)
      CALL SIMP(CZ,XBB,A,NBT,IRR)
      CALL SIMP(CY, XBB, SD, NBT, IR)
      CM=-CZ
      IF(IRR.NE.1) GO TO 1570
      MM = 0
      DO 1460 N = NS1,NS2
      MM = MM + 1
      C(MM) = ZDELTA(N)*C(MM)
      G(MM) = XB(N) + XN - XP + R(N)*FD(N)
      TC(MM) = SD(MM) * ( XB(N) + XN - XP )
      IF (N .EQ. 1) G(MM) = XB(N) + XN    - XP + DS1/(2.*PI)
      IF (N.EQ.NBODYS.AND.R(NBODYS).EQ.0.) G(MM)=XB(N)+XN-XP+DS2/(2.*PI)
 1460 A(MM) = A(MM)*G(MM) + C(MM) + R(N)*DZDX(N)*D(MM)
      CALL SIMP(CM,XBB,A,NBT,IRR)
      CALL SIMP(BBCMY, XBB, TC, NBT, IR)
      IF(IRR.NE.1) GO TO 1570
C
      IF ( YAW .LT. 0.01 .AND. BROLL .LT. 0.01 ) GO TO 1465
      GO TO 1466
 1465 CONTINUE
      CCX = 2.0*CCX
      CY = 0.0
      CZ = 2.0*CZ
 1466 CONTINUE
      CCX = CX
      CCY = -CY
      CCZ = -CZ
      BBCL = -CCX*SINAR*COSB + CCY*(SING*COSAR+COSG*SINB*SINAR) + CCZ*
     1        (COSG*COSAR -SING*SINB*SINAR)
      BBCD = CCX*COSAR*COSB + CCY*(SING*SINAR-COSG*SINB*COSAR) + CCZ*
     1        (SING*SINB*COSAR + COSG*SINAR)
      BBCB = CCX*SINB + CCY*COSG*COSB -CCZ*SING*COSB
      BBCM = CM + CX*(ZN-ZP)
C
      NTHETA = NT
      DO 1670 J=1,NTHETA
 1670 THETAB(J) = THETAB(J) * 57.2957795
      DO 1475 N = 1, NBODYS
 1475 XB(N) = XB(N) + X1
C
 1550 CONTINUE
      WRITE (NTAPEC) NBODYS
      WRITE (NTAPEC) (XB(J),R(J),J=1,NBODYS)
      WRITE (NTAPEC) (T(J),TC(J),J=1,NBODYS)
      WRITE (NTAPEC) T11,TC11
 1560 CONTINUE
C
C -----------------
      RETURN
C -----------------
```

```
C
C
 1570 WRITE(NTAPEO,1690) IRR
      GO TO 1560
 1580 FORMAT (53H****BODY POINT IS OUTSIDE MACH CONE - PROGRAM HALT IN
     118H SUBROUTINE KARMOR)
 1690 FORMAT(1H0,18HINTEGRATION ERROR=I2)
      END
      SUBROUTINE FORMOM(NBODYS,XB,R,XMACH,COSA,SINA,COSB,SINB,UB,VB,VTB
     1         ,COSTB,SINTB,CPBB,I,NTHETA,CPCALC)
C
C     CALCULATION OF PRESSURES, FORCES, AND MOMENTS
C
      DIMENSION XB(1), R(1), UB(1), VB(1), VTB(1), COSTB(1), SINTB(1),
     1          CPBB(51, 1)
C
      BODYL=XB(NBODYS)-XB(1)
      XMACH2 = XMACH**2
      CPSTAG = 1.42857 * ((1.+.2*XMACH2)**3.5 - 1.)/XMACH2
C
      DO 1420 J=1, NTHETA
      VPM = VTB(J) * COSTB(J)  + VB(J) * SINTB(J)
      WPM = VB(J) *  COSTB(J)  - VTB(J) * SINTB(J)
      UWPM  = UB(J)*COSB*COSA - VPM*SINB*COSA + WPM*SINA
C
      UWIND = 1. + UWPM
      VWIND = UB(J)*SINB + VPM*COSB
      WWIND = -UB(J)*COSB*SINA + VPM*SINB*SINA + WPM*COSA
C
      TWIND2 = UB(J)**2 + VPM**2 + WPM**2 - UWPM**2
C
C
      IF ((CPCALC - 1.)  1370,1380,1390
 1370 CPBB(I,J) = -2. * UWPM
      GO TO 1400
 1380 XM2 = XMACH2 - 1.
      CPBB(I,J) = -2.*UWPM + XM2*UWPM**2 - TWIND2
      GO TO 1400
 1390 Q2 = UWIND ** 2   + VWIND **2  + WWIND ** 2
      CPBB(I,J) = 1.42857*((1.+.2*XMACH2*(1.-Q2))**3.5-1.)/XMACH2
 1400 CONTINUE
 1410 CONTINUE
      IF(R(NBODYS).EQ.0.) CPBB(NBODYS,J)=CPSTAG
 1420 CONTINUE
C
      RETURN
      END
      SUBROUTINE CUBERT
      DIMENSION X(3)
      COMMON/LOCTN / V(3), L, M, G(4), VMAX, VMIN
      COMMON/POINTS/ NPTS, NVAR
      L=0
      SUM = SQRT( G(1)**2+G(2)**2+G(3)**2+G(4)**2 )
      G1 = G(1) / SUM
      G2 = G(2) / SUM
      G3 = G(3) / SUM
      G4 = G(4) / SUM
C
C        EQ. 1   G1*X**3 + G2*X**2 + G3*X = 0.
      IF ( ABS( G4) .LT. 1.E-6) GO TO 4
C
C        EQ. 2   G1*X**3 + G2*X**2 + G3*X +G4 = 0.
      IF ( ABS( G1) .GT. 1.E-6) GO TO 9
C
C        EQ. 3   G2*X**2 + G3*X + G4 = 0.
```

```
      IF ( ABS( G2) .GT. 1.E-6) GO TO 7
C
C        EQ. 4   G3*X + G4 = 0.
      IF ( ABS( G3) .GT. 1.E-6) GO TO 5
      RETURN
C
  4   CONTINUE
      V(1) = 0.0
      L=1
      RETURN
  5   CONTINUE
      VO    = - G(4) / G(3)
      GO TO 35
  7   CONTINUE
      R1 = G(3)**2 - 4.0*G(2)*G(4)
      IF (  R1  .GT. 0.0 )  GO TO 107
      WRITE (6,901)
  901 FORMAT (1H0,*NAGATIVE RADICAL RESULTED FROM SOLVING THE QUADRATIC
     1EQUATION  G2.X2 + G3.X + G4 = 0*)
      RETURN
  107 CONTINUE
      R1=SQRT(R1)
      V1 = ( -G(3) + R1 ) *0.5 / G(2)
      V(1) = V1
      IF ( V1  .GT. VMIN .AND.  V1  .LT. VMAX ) GO TO 8
      V2 = (-G(3) - R1 )*0.5 / G(2)
      V(2) = V2
      IF ( V2  .GT. VMIN .AND.  V2  .LT. VMAX ) GO TO 8
      WRITE (6,902) VMIN, V1,V2, VMAX
  902 FORMAT (1H0,*BOTH ROOTS V1, V2 OF QUADRATIC EQ. G2.X2 + G3.X + G4
     1= 0 OUTSIDE OF LIMIT VMIN, VMAX*/ 1X,*VMIN =*,E14.6,4X,*V1 =*,
     2           E14.4,4X,*V2 =*,E14.4,4X,*VMAX =*,E14.4)
      RETURN
  8   CONTINUE
      L = 1
      RETURN
C
C        THREE ROOTS
C
  9   CONTINUE
      A1 = G(2) / G(1)
      A2=G(3)/G(1)
      A3=G(4)/G(1)
      A=A2-A1*A1/3.
      Q=2.*A1*A1*A1/27.-A1*A2/3.+A3
      RAD=Q*Q/4.+A*A*A/27.
      SIGN=1.
      IF(Q.GE.0.)SIGN=-1.
      IF(RAD)10,20,20
C
C              THREE UNEQUAL REAL ROOTS
C
  10  CONTINUE
      A3 = SQRT( ABS( A/3.0) )
      A2 = SQRT( ABS( A**3/3. ) )
      PHI = ACOS( ABS( 1.5*Q/ A2) )
      DO 11 I=1,3
      X(I) = 2.0*SIGN*A3*COS((PHI+6.2831853*FLOAT(I-1))/3.0)
  11  CONTINUE
      GO TO 40
C
C        EITHER ONE REAL AND TWO IMAG. OR THREE REAL ROOTS OF WHICH TWO
C        ARE EQUAL
C
  20  CONTINUE
```

```
      V1 = 1.0 / 3.0
      Q = A2*V1 - ( A1*V1)**2
      R = .5*A1*A2*V1 - .5 *A3 - ( A1*V1) **3
      RAD = Q**3 + R*R
      IF ( RAD  .GT.  0.0 )  GO TO 120
      WRITE (6,903)
  903 FORMAT (1H0,*ERROR IN TESTING RADICAL, REDUNDENT TEST*)
      RETURN
  120 CONTINUE
      RAD = SQRT( RAD)
      A2 = R + RAD
      SIGN = 1.
      IF ( A2  .LT.  0.0)  SIGN = -1.
      A2 = SIGN * ABS( A2 ) **V1
      A3 = R - RAD
      SIGN = 1.
      IF ( A3 .LT. 0.0 ) SIGN = -1.
      A3 = SIGN * ABS( A3 )**V1
      V0 = A2 + A3 - A1*V1
   35 CONTINUE
      IF ( V0 .GT. VMIN .AND. V0  .LT. VMAX ) L = 1
      IF ( L  .EQ. 0 ) WRITE (6,905)
  905 FORMAT (1H0,*THE ONLY UNEQUAL ROOT IS OUTSIDE OF VMIN, VMAX*/1X,
     1         *VMIN =*,E14.4,4X,* V =*,E14.4,4X,*VMAX =*,E14.4)
      V(1) = V0
      RETURN
   40 CONTINUE
      DO 45 I=1,3
      V0 = X(I) - A1/3.0
      X(I) = V0
      V(I) = V0
      IF ( V0 .LT. VMIN .OR.  V0  .GT. VMAX ) GO TO 50
      L=1
      RETURN
   50 CONTINUE
   45 CONTINUE
      WRITE (6,904) VMIN, V(1), V(2), V(3), VMAX
  904 FORMAT (1H0,*ALL THREE UNEQUAL ROOTS OUTSIDE OF LIMIT VMIN, VMAX*/
     1    1X,*VMIN =*,E14.4,4X,*V1 =*,E14.4,4X,*V2 =*,E14.4,4X,*V3 =*,
     2         E14.4,4X,*VMAX =*,E14.4)
C
      RETURN
      END
C
      SUBROUTINE POINT( N )
      COMMON/CURFIT/ B(2,4, 50), T( 50), SCALEF(2)
      COMMON/ ENDPT/ XFIRST, XLAST, XSAVE( 50)
      COMMON/LOCTN/ V(3), L, M, A(4), VMAX, VMIN
      COMMON/POINTS/ NPTS, NVAR
      COMMON/ POINTP/  X(4),  XP( 4),  XPP( 4)
      DIMENSION DX(4)
C
      N1 = NPTS - 1
      IF(N.NE.0)GO TO 4
      DO 1 I=1,N1
      M=I
      IF ( V(1)  .LE. T(I+1) ) GO TO 2
    1 CONTINUE
      WRITE(6,100)
  100 FORMAT(1H0,33H ERROR - PARAMETER EXCEEDS LIMITS)
      RETURN
    2 CONTINUE
      DO 300 K=1,3
      P = V(K)
      DX(K) = 1.E10
      IF ( P .LT. VMIN  .OR.  P  .GT. VMAX ) GO TO 300
      DX(K) = ABS((((B(1,1,M)*P + B(1,2,M))*P + B(1,3,M))*P + B(1,4,M))*
     1            SCALEF(1) - XIND )
  300 CONTINUE
      IF ( DX(1)  .GT. DX(2) ) GO TO 302
      IDENT = 1
      IF ( DX(1)  .GT. DX(3) )  IDENT = 3
      GO TO 305
  302 CONTINUE
      IDENT = 2
      IF ( DX(2)  .GT. DX(3) )  IDENT = 3
  305 CONTINUE
      P = V(IDENT)
C
      DO 3   I=1, NVAR
      X(I)=(((B(I,1,M)*P+B(I,2,M))*P+B(I,3,M))*P+B(I,4,M))*SCALEF(I)
      XP(I)=((3.*B(I,1,M)*P+2.*B(I,2,M))*P+B(I,3,M))*SCALEF(I)
    3 XPP(I)=(6.*B(I,1,M)*P+2.*B(I,2,M))*SCALEF(I)
C
  310 CONTINUE
      RETURN
C
    4 X(N)=X(N)/SCALEF(N)
      DO 5 I=1,N1
      M=I
      K = I + 1
      IF ( XSAVE(K)  .GE. X(N) ) GO TO 6
    5 CONTINUE
      XIND = X(N) * SCALEF(N)
      XFT = XFIRST * SCALEF(1)
      XLT = XLAST * SCALEF(1)
      WRITE (6,101)  XIND, XFT, XLT
  101 FORMAT (1H0,*  THE GIVEN VALUE OF INDEPENDENT VARIABLE *,E12.4,
     1           5X,*IS OUT OF RANGE TO BE INTERPOLATED  *,E12.4,5X,E12.4)
      RETURN
    6 DO 7 I=1,4
    7 A(I)=B(N,I,M)
      A(4)=A(4)-X(N)
      IF ( XFIRST  .EQ. X(N) ) GO TO 112
      IF ( XLAST   .EQ. X(1) ) GO TO 111
      VMAX = T(M+1) * 1.0001
      VMIN = T(M) * 0.9999
      CALL CUBERT
      XIND = X(1)* SCALEF(1)
      GO TO 103
  112 V(1) = 0.0
      L=1
      GO TO 103
  111 V(1) = 1.0
      L=1
  103 CONTINUE
      IF(L.NE.0)GO TO 2
      OP = P
      WRITE (6,10) OP
   10 FORMAT (1X4HOP =,E15.8)
      DO 8 J=1,2
      K = M + J - 1
      B1 = B(N,1,K)
      B2 = B(N,2,K)
      B3 = B(N,3,K)
      B4 = B(N,4,K)
      P = T(K)
      SUM = B1*P**3 + B2*P**2 + B3*P + B4
      DIF = SUM - X(N)
    8 WRITE (6,9) K,P,B1,B2,B3,B4,X(N),SUM,DIF
    9 FORMAT (1XI3,8E15.8)
      WRITE(6,102)
  102 FORMAT(1H0,32H FAILURE IN CUBE ROOT EXTRACTION)
      RETURN
      END
C
      SUBROUTINE SPLFIT
      REAL L,M,MU
      COMMON/CURFIT/ B(2,4, 50), T( 50), SCALEF(2)
      COMMON/ ENDPT/ XFIRST, XLAST, XSAVE( 50)
      COMMON/INPUT /  X( 2,  50)
      COMMON/POINTS/ NPTS, NVAR
      DIMENSION M(2, 50), S( 50), L( 50), MU( 50), D( 50), P( 50), Q( 50
     1          ), U( 50)
      IF ( NPTS .GT.  50) NPTS= 50
      DO 200 I=1, NVAR
  200 SCALEF(I)=0.
      DO 201 I=1,NPTS
      DO 201 J=1,NVAR
  201 SCALEF(J)=AMAX1(SCALEF(J),ABS(X(J,I)))
      DO 101 J=1,NVAR
      IF ( SCALEF(J)  .LT. 1.E-10 ) SCALEF(J) = 1.0
  101 CONTINUE
      DO 202 I=1,NPTS
      DO 202 J=1,NVAR
  202 X(J,I)=X(J,I)/SCALEF(J)
      DO 203 I=1,NPTS
      XSAVE(I) = X(1,I)
  203  CONTINUE
      XFIRST = X(1,1)
      XLAST = X(1,NPTS)
      T(1)=0.
      S(1)=0.
      SUM=0.
      L(1)=-2.
      MU(1)=0.
      L(NPTS)=0.
      MU(NPTS)=-2.
      D(1)=0.
      D(NPTS)=0.
      P(1)=2.
      Q(1)=1.
      U(1)=0.
      N1=NPTS-1
      DO 2 I=2,NPTS
      SUM1=0.
      DO 1 J=1,NVAR
    1 SUM1=SUM1+(X(J,I)-X(J,I-1))**2
      S(I)=SQRT(SUM1)
    2 T(I)=T(I-1)+S(I)
      DO 3 I=2,NPTS
      S(I)=S(I)/T(NPTS)
    3 T(I)=T(I)/T(NPTS)
      DO 4 I=2,N1
      L(I)=S(I+1)/(S(I)+S(I+1))
      MU(I)=1.-L(I)
      P(I)=MU(I)*Q(I-1)+2.
    4 Q(I)=-L(I)/P(I)
      P(NPTS)=2.*(1.-Q(N1))
      DO 7 I=1,NVAR
      DO 5 J=2,N1
      D(J)=6.*(((X(I,J+1)-X(I,J))/S(J+1)-(X(I,J)-X(I,J-1))/S(J))/(S(J)+
     1S(J+1)))
    5 U(J)=(D(J)-MU(J)*U(J-1))/P(J)
      U(NPTS)=-MU(NPTS)*U(N1)/P(NPTS)
      M(I,NPTS)=U(NPTS)
      DO 6 J=1,N1
      K=NPTS-J
      M(I,K)=Q(K)*M(I,K +1)+U(K)
      B(I,1,K)=(M(I,K +1)-M(I,K))/(6.*S(K+1))
      B(I,2,K)=.5*(M(I,K)*T(K+1)-M(I,K+1)*T(K))/S(K+1)
      B(I,3,K)=(.5*(M(I,K+1)*T(K)**2-M(I,K)*T(K+1)**2)+X(I,K+1)-X(I,K)+
     1(M(I,K)-M(I,K+1))*S(K+1)**2/6.)/S(K+1)
    6 B(I,4,K)=(M(I,K)*T(K+1)**3-M(I,K+1)*T(K)**3)/(6.*S(K+1))+(X(I,K)*T
     1(K+1)-X(I,K+1)*T(K))/S(K+1)+S(K+1)/6.*(M(I,K+1)*T(K)-M(I,K)*T(K+1)
     1)
    7 CONTINUE
      RETURN
      END
      OVERLAY(WANG, 7, 7)
      PROGRAM FCALC
C
C ..................................................................
C     CONTROL SUBROUTINE FOR CALCULATION OF FORCES, MOMENTS
C     AND PRESSURES
C ..................................................................
C
      COMMON/ MAIN / NTAPEA,NTAPEB,NTAPEC,NTAPED,NTAPEE,NTAPEF,NTAPEI,
     1         NTAPEO,NBODY,NWING,XMACH,SYM,KACE,NPOLAR,IRW
      COMMON /AVAR/  A(210),ACB(21),ABX(100),AWX(110),AREA(210),ARN(2),A
     1               RWT(20),ALPHA(210),ALPHAB(210),ALPHAT(110),
     2               ALPHAW(110),ALPHAX(110),AWS(110),ABT(100),AMIN(
     3               210,2),ALPHAA,ALPHAD,ARA,ARADEG,ARD,ARW(10), ARAS,
     4               AT, AAA
      COMMON /BVAR/  B(210), BBCL, BBCD, BBCY, BBCM, BBCMR, BBCMY
C
      COMMON/BODYSP/ DRDX(51), DZDX(51), DRDXC(51)
      COMMON/CVAR/C(210), CHORD(210),CL(210),CPNN(1,1,1),CLS(210)
     1               ,CDR,CASE,CPCALC,CBAR,CONSNT,CLBAR,CLX,CLM,CDN
      COMMON/DVAR/ D(210),DZDXB(51), DADEG,DARAD
      COMMON /IVAR/  IPOLAR
      COMMON /KVAR/  KASE,KONFIG,KPOLAR
      COMMON /NVAR/  NFMT(9),NROW(2)
     1         ,NS,NPANEL,NACEL,NROWB,NROWW,NCOLW,NTHETA,NTHETS,
     2         NXLE, NRG, NPOLR, NCLX
      COMMON /PVAR/  POLAR
      COMMON/RVAR/ R(51), RN(51,2), RATIOX, RFAREA
      COMMON /SVAR/  SEMIS,SLC
      COMMON/TVAR/ TITLE(20), THETA(210), THETAB(15),THETAS(15), THETAN
     1         (11,2), TCL(11), TCD(11), TCM(11), THICK,TWIST
      COMMON /UVAR/  UBWT(150),UWBT(110),UWWT(110),UNCL(1,1)
      COMMON /VVAR/  VBWT(150),VWBT(110),VWWT(110),VNCL(1,1)
     1               ,VOUT
      COMMON /WVAR/  WBWT(150),WWBT(110),WWWT(110),WNCL(1,1)
      COMMON/XVAR/ XBAR(210), XC(210), XMI(2), XMXN(2), XMTN(2), XB(51),
     1         XN(51,2), XNNCD(2), XNNCL(2), XNNCM(2), XYZ(3), XCL(11),
     2         XCD(11), XNACEL, XP, XCPBAR
      COMMON /YVAR/  YBAR(210),YC(210),YMI(2)
      COMMON/ZVAR/ ZBAR(210), ZC(210), ZMI(2), ZDELTA(51), ZBN(51,2)
     1               ,ZP,ZA
      COMMON /PANVEL/ VS1(110),VS2(110),WSS(110)
      COMMON/PBDATA/ NBODYS, XMIDB(51), NTHETB, THETB(15), CPBB(51,15)
     1         ,XCB(3,700),NG(350),NELM(700),NE,NGRIDP,PINF,IPUNCH, CPB
      COMMON/WINGOT/ CPU, CPL,NWG,ISOLID,IFORM(10),TAU,DROLL
      COMMON/REPEAT/ IREPT
C
      DIMENSION  VVX(110),WVX(110),VWX(110),WWX(110)
      DIMENSION VSS(110),WAS(110)
      DIMENSION ALPHAL(110),ALPHAU(110)
      DIMENSION CHORDL(20), CLBB(110),CLWW(110),CPB(150),CPU(110),
     1          CPL(110), CSHAPE(110)
      DIMENSION U(150),UBB(150),UBN(100),UBW(150),UWB(110),UWN(110),UWW(
     1          110)
      DIMENSION V(150),VBB(150),VBN(100),VBW(150),VWB(110),VWN(110),VWW(
     1          110)
      DIMENSION W(150),WBB(150),WBN(100),WBW(150),WWB(110),WWN(110),WWW(
     1          110),WSCLL(20),WSCLU(20),WSCDL(20),WSCDU(20),WSCL(20),
     2          WSCD(20),WSCBU(20),WSCBL(20),WSCB(20),CF(140)
C
      INTEGER TITLE
C
C
C ----------------------------------------------------------------------
C     KONFIG = 1     WING ALONE CONFIGURATION
C     KONFIG = 2     BODY ALONE CONFIGURATION
C     KONFIG = 3     WING-BODY CONFIGURATION
C ----------------------------------------------------------------------
C
      PI = 3.1415926535898
      BETA=SQRT(ABS(XMACH*XMACH-1.))
      BETA2=BETA*BETA
      BETAM=2.4/1.414214*XMACH**4/BETA**.75
      ARAXX=ARA
      ARA=0.0
      RATIOD=.75
      IF(KONFIG.EQ.1) NR=NROW(1)
      IF(KONFIG.EQ.3) NR=NROW(2)
C
C     ANY CONFIGURATION
      IF (KONFIG-2)1050,1000,1010
C
C     BODY-ALONE CONFIGURATION
 1000 CONTINUE
      CALL FSF(10,NTAPEC,IRR)
      CALL FOR EOF (NTAPEC)
      REWIND NTAPEC
      GO TO 1530
 1010 CONTINUE
C
C     WING ALONE OR WING-BODY CONFIGURATION
 1050 CONTINUE
 1120 CONTINUE
C
C     CALCULATE WING CL, GIVEN WING ALPHA     KASE = 2
 1180 CONTINUE
      IF (KONFIG-2)1190,1210,1210
C
C     WING ALONE CONFIGURATION
 1190 CONTINUE
      DO 1200 J=1,NWING
 1200 AWX(J)=ALPHAW(J)-ALPHAX(J)
      CALL FSF (1,NTAPEE,IRR)
      CALL DCPD(NWING,NTAPEE,A,AWX,CL)
      REWIND NTAPEE
      GO TO 1320
C
C     WING-BODY CONFIGURATION
 1210 CONTINUE
      DO 1220 J=1,NWING
 1220 AWX(J)=ALPHAW(J)-ALPHAX(J)
      DO 1230 J=1,NBODY
      READ (NTAPEF) (A(I),I=1,NWING)
      DO 1230 I=1,NWING
 1230 AWX(I)=AWX(I)-A(I)*ABX(J)
      CALL FSF (1,NTAPEE,IRR)
      CALL DCPD (NWING,NTAPEE,A,ANX,CL(NS))
      REWIND NTAPEE
      CALL FSF (1,NTAPEF,IRR)
      CALL DCPI (NBODY,NWING,NTAPEF,A,ABX,CL(NS),CL)
      REWIND NTAPEF
C     WING ALONE OR WING-BODY CONFIGURATION
 1320 CONTINUE
      IF (KONFIG-2)1330,1340,1340
C
C     WING ALONE CONFIGURATION
 1330 CONTINUE
      CALL FSF (1,NTAPED,IRR)
      CALL CVEL (NWING,NWING,NTAPED,A,B,C,CL,UWW,VWW,WWW)
      REWIND NTAPED
      GO TO 1350
C
C     WING-BODY CONFIGURATION
 1340 CONTINUE
      CALL FSF (1,NTAPED,IRR)
      CALL CVEL (NWING,NBODY,NTAPED,A,B,C,CL,UWB,VWB,WWB)
      CALL FSF (2,NTAPED,IRR)
      CALL CVEL (NWING,NWING,NTAPED,A,B,C,CL(NS),UWW,VWW,WWW)
      REWIND NTAPED
C
C     WING ALONE OR WING-BODY CONFIGURATION
 1350 CONTINUE
      CALL FSF(10,NTAPEC,IRR)
      WRITE (NTAPEC) NPANEL
      WRITE (NTAPEC) (CL(J),J=1,NPANEL)
      CALL FOR EOF (NTAPEC)
      REWIND NTAPEC
      DO 1345 J=1,NWING
      JJ=J+NBODY
      NJ=(J-1)/NR+1
      VSS(J)=VS1(J)*CL(JJ)
      DO 1345 I=1,NWING
      II=I+NBODY
      NI=(I-1)/NR+1
      IF(I.LT.J.AND.NI.EQ.NJ) VSS(J)=VSS(J)+VS2(I)*CL(II)
 1345 CONTINUE
      NWPW = NWING / NWG
      DO 1360 J=1,NWING
      JWS = (J-1)/NWPW + 1
      JJ=J+NBODY
      TJ=THETA(JJ)
      COSTJ=COS(TJ)
      SINTJ=SIN(TJ)
      VVX(J)=VSS(J)*COSTJ
      WVX(J)=VSS(J)*SINTJ
      IF(THICK.EQ.0.) GO TO 1360
      WWX(J)=WSS(J)*COSTJ
      VWX(J)=WSS(J)*SINTJ
 1360 UWW(J)=UWW(J)-.25*CL(JJ)
C
      IF (THICK)1370,1410,1370
 1370 CONTINUE
      SINB = SIN(BROLL)
      COSB = COS(BROLL)
      DO 1380 J=1,NWING
      U(J)=UWB(J)+UWW(J)+UWBT(J)+UWWT(J)+UWN(J)
      VJ =VWB(J)+VWW(J)+VWBT(J)+VWWT(J)+VWN(J)+VWX(J)+VVX(J)
      WJ =WWB(J)+WWW(J)+WWBT(J)+WWWT(J)+WWN(J)+WWX(J)+WVX(J)
      V(J) = VJ*COSB - WJ*SINB
      W(J) = VJ*SINB + WJ*COSB
 1380 CONTINUE
```

```
      ARA = ARAXX
      CALL  CP(NWING,XMACH,CPCALC,U,V,W,CPU,ARA,YAW)
      DO 1390 J=1,NWING
      JJ=J+NBODY
      U(J)=U(J)+.5*CL(JJ)
      VJ =VWB(J)+VWW(J)+VWBT(J)+VWWT(J)+VWN(J)+VWX(J)+VVX(J)
      WJ =WWB(J)+WWW(J)+WWBT(J)+WWWT(J)+WWN(J)+WWX(J)-WVX(J)
      V(J) = VJ*COSB - WJ*SINB
      W(J) = VJ*SINB + WJ*COSB
 1390 CONTINUE
      CALL  CP(NWING,XMACH,CPCALC,U,V,W,CPL,ARA,YAW)
      DO 1400 J=1,NWING
      ALPHAU(J)=ALPHAW(J)+WSS(J)
 1400 ALPHAL(J)=ALPHAW(J)-WSS(J)
      GO TO 1450
C
C     WING ALONE OR WING-BODY CONFIGURATION
C     WITHOUT WING THICKNESS
 1410 CONTINUE
      DO 1420 J=1,NWING
      U(J)=UWB(J)+UWW(J)+UWBT(J)+UWN(J)
      V(J)=VWB(J)+VWW(J)+VWBT(J)+VWN(J)+VVX(J)
 1420 W(J)=WWB(J)+WWW(J)+WWBT(J)+WWN(J)+WVX(J)
      CALL  CP(NWING,XMACH,CPCALC,U,V,W,CPU,ARA,YAW)
      DO 1430 J=1,NWING
      JJ=J+NBODY
      V(J)=VWB(J)+VWW(J)+VWBT(J)+VWN(J)-VVX(J)
      W(J)=WWB(J)+WWW(J)+WWBT(J)+WWN(J)-WVX(J)
 1430 U(J)=U(J)+.5*CL(JJ)
      CALL  CP(NWING,XMACH,CPCALC,U,V,W,CPL,ARA,YAW)
      DO 1440 J=1,NWING
      ALPHAU(J)=ALPHAW(J)
 1440 ALPHAL(J)=ALPHAW(J)
C
C     WING ALONE OR WING-BODY CONFIGURATION
C     CALCULATE CAMBER SHAPE
C
 1450 CONTINUE
C
C
      DO 1451 J=1,NWING
      JJ=1+(J-1)/NROWW
      WAS(J)=ARWT(JJ)/CDR
 1451  WAS(J)=COS(WAS(J))*COS(ARAXX)
C
      CALL CAMBER(NCOLW,NROWW,CHORD(NS),CHORDL)
C
C     COMPUTE COEFFICIENTS OF LIFT, DRAG, MOMENT ON WING
      CALL WLDM(NWING,NROWW,XP,ZP,RFAREA,AREA(NS),XBAR(NS),YBAR(NS),
     1         ZBAR(NS),ALPHAU,THETA(NS),CPU,WSCLU,WSCDU,WSCBU,WCLU,
     2         WCDU,WCBU,WCMPU,WCMRU,WCMYU,ARA)
      CALL WLDM(NWING,NROWW,XP,ZP,RFAREA,AREA(NS),XBAR(NS),YBAR(NS),
     1         ZBAR(NS),ALPHAL,THETA(NS),CPL,WSCLL,WSCDL,WSCBL,WCLL,
     2         WCDL,WCBL,WCMPL,WCMRL,WCMY_,ARA)
      DO 1460 J=1,NCOLW
      JN=(J-1)*NROWW+(NROWW+1)/2
      JN=NBODY+JN
      DY=AREA(JN)/CHORD(JN)
      WSCD(J)=WSCDL(J)-WSCDU(J)
      WSCL(J)= WSCLL(J)-WSCLU(J)
      WSCB(J) = WSCBL(J) - WSCBU(J)
 1460 CONTINUE
      WCL = WCLL - WCLU
      WCD = WCDL - WCDU
      WCB = WCBL - WCBU
 1465 CONTINUE
      WCPM = (WCMPL - WCMPU) / CBAR
      WCRM = (WCMRL - WCMRU) / CBAR
      WCYM = (WCMYL - WCMYU) / CBAR
      IF (KONFIG .EQ. 1) GO TO 1540
C
C     WING-BODY CONFIGURATION
      CALL CVEL (NBODY,NBODY,NTAPED,A,B,C,CL,UBB,VBB,WBB)
      CALL FSF (2,NTAPED,IRR)
      CALL CVEL (NBODY,NWING,NTAPED,A,B,C,CL(NS),UBW,VBW,WBW)
      REWIND NTAPED
      DO 1470 J=1,NBODY
 1470 UBB(J)=UBB(J)-.25*CL(J)
C
C     COMPUTE COEFFICIENT OF PRESSURE ON BODY WITH OR WITHOUT THE
C     EFFECT OF WING THICKNESS
C
      NBD = NBODY
      NRB = NROWB
      NTA = NTHETA
      IF ( BROLL .LT. 0.01 .AND. YAW .LT. 0.01 ) GO TO 1488
      NTA = 2*NTHETA
      NBD = 2 * NBODY
      NRB = 2 * NROWB
C
      DO 1485 J =1, NTHETA
      THETAB(J+NTHETA) = THETAB(J) + 180.0
 1485 CONTINUE
 1488 CONTINUE
C
 3300 CONTINUE
C
C
      DO 1490 J = 1, NROWB
      DO 1490 I = 1, NTHETA
      IJ = J + (I-1)*NROWB
      U(IJ) = UBB(IJ) + UBW(IJ) + UBWT(IJ)
      V(IJ) = VBB(IJ) + VBW(IJ) + VBWT(IJ)
      W(IJ) = WBB(IJ) + WBW(IJ) + WBWT(IJ)
 1490 CONTINUE
C
      CALL  CP(NBD,XMACH,CPCALC,U,V,W,CPB,ARA,YAW)
C
C     COMPUTE COEFFICIENTS OF LIFT, DRAG, MOMENT ON BODY PANELS
C
      IF ( BROLL .LT. 0.01 .AND. YAW .LT. 0.01) GO TO 1520
      DO 3000 I=1, NTHETA
      IR = NTHETA - I + 1
      IR1 = (IR-1)*NROWB
      IR2 = (I + NTHETA - 1)*NROWB
      DO 3000 J=1, NROWB
      IJ = IR1 + J
      JI = IR2 + J
      CPB(JI) = CPB(IJ)
 3000 CONTINUE
C
 1520 CONTINUE
C
      CALL BLDM(ARA,YAW,BROLL,BRDXC,AREA,CPB,XP,XBAR,NROWB,
     1  NTHETA,NTA,THETAB,RFAREA,CBAR,BWCL,BWCD,BWCB,BWCM,BWCYM)
C
C     BODY ALONE OR WING-BODY CONFIGURATION
 1530 CONTINUE
      BBCL=BBCL/RFAREA
      BBCD=BBCD/RFAREA
      BBCY = BBCY / RFAREA
      BBCM=BBCM/(RFAREA*CBAR)
      BBCYM = BBCMY/(RFAREA*CBAR)
C
      IF (KONFIG .EQ. 2) GO TO 1540
C
C         WING-BODY CONFIGURATION
C         BODY COEFFICIENTS
C
      BWCM=BWCM/CBAR
      BCL=BWCL+BBCL
      BCD=BWCD+BBCD
      BCY = BWCB
      BCM=BWCM+BBCM
      BCYM = BBCYM + BWCYM
C
C         MISSILE COEFFICIENTS
C
      WBCL=BCL+WCL+XNNCLT
      WBCD=BCD+WCD+XNNCDT
      WBCY = BCY + WCB
      WBCM=BCM+WCPM+XNNCMT
      WBCRM = WCRM
      WBCYM = BCYM + WCYM
C .
C     ANY CONFIGURATION
C     WRITE OUTPUT
 1540 CONTINUE
      ARA=ARAXX
C
      IF (KONFIG-2)1630,1560,1550
C
C     WING-BODY CONFIGURATION
 1550 CONTINUE
      WRITE (NTAPEO,1770)
      GO TO 1570
C
C     BODY ALONE CONFIGURATION
 1560 CONTINUE
      WRITE (NTAPEO,1820)
C
C     BODY ALONE OR WING-BODY CONFIGURATION
 1570 CONTINUE
      WRITE (NTAPEO,2010) BBCD,BBCL,BBCY,BBCM,BBCRM,BBCYM
      WRITE (NTAPEO,1830)
      WRITE(NTAPEO,1860) (THETB(I),I=1,NTHETB)
      WRITE (NTAPEO,1870)
      IF(XMACH.LT.1.) GO TO 1590
      NBS = NBODYS
      DO 1580 I=1,NBODYS
      XMIDB(I) = XB(I)
 1580 WRITE (NT*PEO,1880)XB(I),(CPBB(I,J),J=1,NTHETB)
      GO TO 1610
 1590 NBS=NBODYS-1
      DO 1600 I=1,NBS
      XX=.5*(XB(I)+XB(I+1))
      XMIDB(I) = XX
 1600 WRITE (NTAPEO,1880) XX, (CPBB(I,J), J=1,NTHETB)
 1610 IF(KONFIG.EQ.2) GO TO 1750
C
C     WING-BODY CONFIGURATION
      WRITE (NTAPEO,1790)
      WRITE (NTAPEO,2010) BWCD,BWCL,BWCB,BWCM,BWCRM,BWCYM
      WRITE (NTAPEO,1840)
      CALL OUTB (NTAPEO,NBB,NTA,NROWB,THETAB,CPB)
      WRITE (NTAPEO,1850)
      CALL OUTB (NTAPEO,NBD,NTA,NROWB,THETAB,ALPHAB)
C
C     WING ALONE OR WING-BODY CONFIGURATION
 1630 CONTINUE
C
      IF (NACEL .EQ. 0) GO TO 1680
C
C     WING ALONE OR WING-BODY CONFIGURATION WITH NACELLE(S)
      DO 1670 J=1,NACEL
      NTN=XNTN(J)
      NXN=XNXN(J)
      WRITE (NTAPEO,1890) J
      WRITE (NTAPEO,2010) XNNCD(J),XNNCL(J),XNNCM(J)
      WRITE (NTAPEO,1900)
      WRITE (NTAPEO,1860) (THETAN(I,J),I=1,NTN)
      WRITE (NTAPEO,1870)
      IF(XMACH.LT.1.) GO TO 1650
      DO 1640 I=1,NXN
 1640 WRITE (NTAPEO,1880) XN(I,J),(CPNN(I,K,J),K=1,NTN)
      GO TO 1670
 1650 NXS=NXN-1
      DO 1660 I=1,NXS
      XX=.5*(XN(I,J)+XN(I+1,J))
 1660 WRITE(NTAPEO,1880)   XX, (CPNN(I,K,J),K=1,NTN)
 1670 CONTINUE
C
 1680 CONTINUE
      DO 1690 J=NS,NPANEL
 1690 CL(J)=-CL(J)
C
      IF (KONFIG-2)1700,1710,1710
C
C     WING ALONE CONFIGURATION
 1700 CONTINUE
      WRITE (NTAPEO,1810)
      GO TO 1720
C
C     WING-BODY CONFIGURATION
 1710 CONTINUE
      WRITE (NTAPEO,1780)
C
C     WING ALONE OR WING-BODY CONFIGURATION
 1720 CONTINUE
      WRITE (NTAPEO,2010) WCD, WCL, WCB, WCPM, WCRM, WCTM
      WRITE (NTAPEO,1980)
      WRITE (NTAPEO,2000) (I,I=1,NCOLW)
      WRITE (NTAPEO,1760) (WSCB(I),I=1,NCOLW)
      WRITE (NTAPEO,1990)
      WRITE (NTAPEO,2000) (I,I=1,NCOLW)
      WRITE (NTAPEO,1760) (WSCL(I),I=1,NCOLW)
      WRITE (6, 1991)
      WRITE (6, 2000) ( I, I=1,NCOLW)
      WRITE (6, 1760) ( WSCB(I),I=1,NCOLW)
      WRITE (NTAPEO,1910)
      CALL OUTW (NTAPEO,NWING,NCOLW,NROWW,CPU)
      WRITE (NTAPEO,1920)
      CALL OUTW (NTAPEO,NWING,NCOLW,NROWW,CPL)
      WRITE (NTAPEO,1930)
      CALL OUTW (NTAPEO,NWING,NCOLW,NROWW,CL(NS))
      IF (THICK)1730,1740,1730
 1730 WRITE (NTAPEO,1950)
      CALL OUTW (NTAPEO,NWING,NCOLW,NROWW,ALPHAU)
      WRITE (NTAPEO,1960)
      CALL OUTW (NTAPEO,NWING,NCOLW,NROWW,ALPHAL)
      WRITE (NTAPEO, 970)
      CALL OUTW (NTAPEO,NWING,NCOLW,NROWW,ALPHAW)
      WRITE (NTAPEO,1940)
      WRITE (NTAPEO,1760)(CHORDL(I),I=1,NCOLW)
C
      IF (KONFIG .EQ. 1) GO TO 1750
C
C     WING-BODY CONFIGURATION
      WRITE (NTAPEO,1800)
      WRITE (NTAPEO,2010) WBCD,WBCL,WBY, WBCM, WBCRM, WBCYM
C
C     ANY CONFIGURATION
 1750 CONTINUE
C
C
C                 REDUCE POLAR BY 1 FOR EACH PASS
C
      POLAR = POLAR - 1
C -----------------
C -----------------
C
C
C     FORMAT STATEMENTS
 1760 FORMAT(1H ,F29.5,9F10.5)
 1770 FORMAT(1H1, 48H PRESSURES, FORCES, AND MOMENTS ON ISOLATED BODY)
 1780 FORMAT(1H1, 66H PRESSURES, FORCES, AND MOMENTS ON WING PANELS IN P
     1RESENCE OF BODY)
 1790 FORMAT(1H1, 70H INCREMENTAL PRESSURES, FORCES, AND MOMENTS ON BODY
     1 PANELS DUE TO WING)
 1800 FORMAT(1H1, 44H FORCES AND MOMENTS ON WING-BODY COMBINATION)
 1810 FORMAT(1H1, 38H PRESSURES, FORCES AND MOMENTS ON WING)
 1820 FORMAT(1H1, 38H PRESSURES, FORCES AND MOMENTS ON BODY)
 1830 FORMAT(1H0/32H  BODY PRESSURE COEFFICIENTS(CP))
 1840 FORMAT(1H0/37H  BODY PANEL PRESSURE COEFFICIENT(CP))
 1850 FORMAT(1H0/25H  BODY PANEL SLOPE(DR/DX))
 1860 FORMAT(1H0,1X,11HTHETA(DEG.),12F10.2/(13X,12F10.4) )
 1870 FORMAT(1H ,9X,1HX)
 1880 FORMAT(1H ,F11.4,12F10.4/(13X,12F10.4) )
 1890 FORMAT(1H1, 57H PRESSURES, FORCES, AND MOMENTS ON ISOLATED NACELLE
     1 PAIR ,I2)
 1900 FORMAT(1H0/34H NACELLE PRESSURE COEFFICIENTS(CP))
 1910 FORMAT(1H0/52H  UPPER SURFACE WING PANEL PRESSURE COEFFICIENTS(CP)
     1)
 1920 FORMAT(1H0/52H  LOWER SURFACE WING PANEL PRESSURE COEFFICIENTS(CP)
     1)
 1930 FORMAT(1H0/36H  WING PANEL PRESSURE DIFFERENCE(CL))
 1940 FORMAT(1H0/23H  WING CHORD LENGTHS(C))
 1950 FORMAT(1H0/39H  UPPER SURFACE WING PANEL SLOPE(DZ/DX))
 1960 FORMAT(1H0/39H  LOWER SURFACE WING PANEL SLOPE(DZ/DX))
 1970 FORMAT(1H0/26H  WING CAMBER SLOPE(DZ/DX))
 1980 FORMAT(1H0/25H  SECTION CD DISTRIBUTION)
 1990 FORMAT(1H0/25H  SECTION CL DISTRIBUTION)
 2000 FORMAT(1H0,2X,16HSPANWISE STATION,I8,9I10,/(18X,I8,9I10))
 2010 FORMAT (1H0/*  CD =*,F9.4,5X,*CL =*,F9.4,5X,*CB =*,F9.4,5X,
     1        *CPM =*,F9.4,5X,*CRM =*,F9.4,5X,*CYM =*,F9.4)
 1991 FORMAT (1H0/2X,*SECTION CD DISTRIBUTION*)
C
      END
C
C/    LIST,NONE
      SUBROUTINE OUTW (NTAPEO,NWING,NCOLW,NROWW,Z)
C
C ..................................................................
C     WING OUPUT FORMAT
C ..................................................................
```

```
C
      DIMENSION Z(1)
C
      WRITE (NTAPEO,1010) (I,I=1,NCOLW)
      WRITE (NTAPEO,1020)
      DO 1000 J=1,NROWW
 1000 WRITE (NTAPEO,1030) (J,(Z(I),I=J,NWING,NROWW))
C
C ------------------
      RETURN
C ------------------
 1010 FORMAT(1H0,2X,16HSPANWISE STATION,I8,9I10/(18X,I8,9I10))
 1020 FORMAT(1H ,2X,17HCHORDWISE STATION)
 1030 FORMAT(1H ,I10,F19.5,9F10.5/(F29.5,9F10.5))
      END
C
      SUBROUTINE OUTB (NTAPEO,NBODY,NTHETA,NROWB,THETA,Z)
C
C ..................................................................
C     BODY OUTPUT FORMAT
C ..................................................................
C
      DIMENSION Z(1),THETA(1)
C
      WRITE (NTAPEO,1010) (THETA(I),I=1,NTHETA)
      WRITE (NTAPEO,1020)
      DO 1000 J=1,NROWB
 1000 WRITE (NTAPEO,1030) (J,(Z(I),I=J,NBODY,NROWB))
C
C ------------------
      RETURN
C ------------------
 1010 FORMAT(1H0,1X,11HTHETA(DEG.),12F10.2/(13X,12F10.4))
 1020 FORMAT(1H0,4X,7HROW NO.)
 1030 FORMAT(1H ,I8,3X,12F10.4/(13X,12F10.4) )
      END
      SUBROUTINE ELDM(ARA,YAW,BROLL,DRDXC,AREA,CPB,XP,XBAR,NROWB,
     1  NTHETA,NTA,THETA,RFAREA,CBAR,BWCL,BWCD,BWCB,BWCM,BWCYN)
C
C  -     COMPUTES LIFT, DRAG, BANK, PITCH AND BANKING MOMENTS FROM
C       THE PRESSURE COEFFICIENT CPB
C
      DIMENSION DRDXC(1), AREA(1), CPB(1), XBAR(1), THETA(1),
     1       COST(20), SINT(20)
C
      TANA = TAN(ARA)
      COSA = COS(ARA)
      SINA = SIN(ARA)
      TANY = TAN(YAW)
      A = 1.0 / SQRT( 1.0 + TANA**2 + TANY**2 )
      COSB = A * SQRT( 1.0 + TANA**2 )
      SINB = SQRT( 1.0 - COSB**2 )
      SING = SIN(BROLL)
      COSG = COS(BROLL)
      IJ = 0
      NBD = NTA*NROWB
C
      DO 300 I=1, NTA
      COST(I) = COS( THETA(I) )
      SINT(I) = SIN( THETA(I) )
C
      DO 250 J=1, NROWB
      IJ = IJ + 1
      JI = IJ
      IF ( I .GT. NTHETA ) JI = (NTA-I)*NROWB + J
      F = CPB(IJ) * AREA(JI)
      FXI = F*DRDXC(J)
      SIND = SQRT( 1.0 - DRDXC(J)**2 )
      FYI = F*SIND*SINT(I)
      FZI = -F*SIND*COST('`
      FX = FX + FXI
      FY = FY + FYI
      FZ = FZ + FZI
      XMP = XMP + FZI*(XBAR(JI) - XP )
      XMY = XMY + FYI*(XBAR(JI) - XP )
  250 CONTINUE
C
      FY = -FY
      FZ = -FZ
  300 CONTINUE
      IF ( NTA .EQ. NTHETA ) FY = 0.0
      BWCL = (-FX*SINA*COSB + FY*(SING*COSA+COSG*SINB*SINA) + FZ*
     1        (COSG*COSA -SING*SINB*SINA) )/RFAREA
      BWCD = (FX*COSA*COSB +  FY*(SING*SINA-COSG*SINB*COSA) +  FZ*
     1        (SING*SINB*COSA + COSG*SINA) )/RFAREA
      BWCB = (FX*SINB + FY*COSG*COSB - FZ*SING*COSB )/RFAREA
      BWCM = XMP / RFAREA / CBAR
      BWCYM = -XMY / RFAREA / CBAR
C
      RETURN
      END
C
      SUBROUTINE WLDM(MM,NROW,XP,ZP,RFAREA,AREA,XBAR,YBAR,ZBAR,ALPHAM,
     1      THETAM,CPM,SCL,SCD,SCB,CL,CD,CB,CMP,CMR,CMY,ARA)
C
C ..................................................................
C     COMPUTES COEFFICIENT OF LIFT, DRAG, AND MOMENT ON WING
C     SPANWISE DISTRIBUTION OF LIFT AND DRAG COEFFICIENTS
C ..................................................................
C
      COMMON/WINGOT/ CPU, CPL,NWG,ISOLID,IFORM(10),YAW,BROLL
      DIMENSION AREA(1),XBAR(1),ZBAR(1),ALPHAM(1),THETAM(1),CPM(1)
     1        ,SCL(1),SCD(1),SCB(1),YBAR(1)
      DIMENSION AWS(1), CPU(110), CPL(110)
C
      PI=3.14159
      NCOL=MM/NROW
      NWC = NCOL / NWG
      WDRAG=0.0
      WLIFT=0.0
      WBANK = WPM = WRM = WYM = 0.0
C
      J=0
      COSY = COS(YAW)
      SINY = SIN(YAW)
      COSA = COS(ARA)
      SINA = SIN(ARA)
      TANA = TAN(ARA)
      TANY = TAN(YAW)
      A = 1.0/SQRT( 1.0 + TANA**2 + TANY**2 )
      COSB = A* SQRT( 1.0 + TANA**2 )
      SINB = SQRT( 1.0 - COSB**2)
      COSG = COS(BROLL)
      SING = SIN(BROLL)
C
      DO 1010 K=1,NCOL
      NWS = (K-1)/NWC + 1
      DRAG=0.0
      XLIFT=0.0
      BANK = PMOM = RMOM = YMOM = 0.0
      SUM=0.0
      CON=1.0
C
      DO 1000 I=1,NROW
      J=J+1
      IF(I.EQ.1) TEST=ABS(YBAR(J))+ABS(ABS(THETAN(J))-.5*PI)
      IF(TEST.LT.0.001) CON=.5
      SUM=AREA(J)+SUM
      SINAP= SIN(ALPHAN(J))
      COSAP= COS(ALPHAN(J))
      SINT = SIN(THETAN(J))
      COST = COS(THETAN(J))
      F=CPN(J)*AREA(J)
      FX = -F*COST*SINAP
      FY = -F*SINT
      FZ = F*COST*COSAP
C
C        SIGN CONVENTIONS ARE FOR POSITIVE PRESSURE ON THE LOWER
C       WING SURFACE, SIGN OF UPPER SURFACE FORCES ARE REVERSED IN
C       THE PROGRAM FCALC
C
      XL = -FX*SINA*COSB + FY*(SING*COSA+COSG*SINB*SINA) + FZ*
     1        (COSG*COSA -SING*SINB*SINA)
      XD = FX*COSA*COSB +  FY*(SING*SINA-COSG*SINB*COSA) +  FZ*
     1        (SING*SINB*COSA + COSG*SINA)
      XB =  FX*SINB + FY*COSG*COSB - FZ*SING*COSB
      BANK = BANK + XB
C
      XLIFT=XLIFT+XL
      DRAG=DRAG+XD
      PMOM = PMOM - XL*(XBAR(J)-XP) + XD*(ZBAR(J)-ZP)
      RMOM = RMOM + F * COSA * SQRT( YBAR(J)**2 + (ZBAR(J)-ZP)**2 )
      YMOM = YMOM + XB *(XBAR(J)-XP) - XD*YBAR(J)
 1000 CONTINUE
C
      SCL(K) = XLIFT / SUM
      SCD(K) = DRAG / SUM
      SCB(K) = BANK / SUM
      WDRAG=WDRAG+DRAG
      WLIFT=WLIFT+XLIFT
      WBANK = WBANK + BANK
      WPM = WPM + PMOM
      WRM = WRM + RMOM
      WYM = WYM + YMOM
 1010 CONTINUE
C
      CL=WLIFT/RFAREA
      CD=WDRAG/RFAREA
      CB = WBANK / RFAREA
      CMP = WPM / RFAREA
      CMR = WRM / RFAREA
      CMY = WYM / RFAREA
C
C -----------------
      RETURN
C -----------------
      END
C
      SUBROUTINE CAMBER(NCOLW,NROWW,CHORD,CHORDL)
C
C ......................................................................
C     COMPUTES SURFACE SHAPE, GIVEN SURFACE SLOPES
C ......................................................................
C
      DIMENSION CHORD(1),CHORDL(1)
```

```
C
      DO 1020 J=1,NCOLW
      CHORDL(J)=0.
      DO 1000 I=1,NROWW
      K=(J-1)*NROWW+I
 1000 CHORDL(J)=CHORDL(J)+CHORD(K)
 1020 CONTINUE
C
C -----------------
      RETURN
C -----------------
      END
C
      SUBROUTINE CP(NP,XMACH,CPCALC,U,V,W,CPP,ARA,YAW)
C
C ..................................................................
C     COMPUTES  LINEAR, NONLINEAR, OR EXACT CP
C ..................................................................
C
      DIMENSION U(1),V(1),W(1),CPP(1)
C
      XM2=XMACH*XMACH
      BT2   =  XM2 - 1.
      TARA = TAN(ARA)
      TYAW = TAN(YAW)
      A = 1.0/SQRT( 1.0 + TARA**2 + TYAW**2 )
      SINA = SQRT( 1.0 - A**2)
      IF(XM2.EQ.0.) GO TO 1000
      Q000  =  1.42857/XM2
      Q001  =  0.2 * XM2
 1000 DO 1050 J=1,NP
      UWPN = U(J)*A + W(J)*SINA
      UWIND = 1. + UWPN
      TWIND2 = V(J)**2 + ( W(J)*A - U(J)*SINA )**2
C
      IF (CPCALC - 1.) 1010,1020,1030
 1010 CPP(J) = - 2. * UWPN
      GO TO 1050
 1020 CPP(J) = -2.*UWPN  + BT2 * UWPN**2 - TWIND2
      GO TO 1050
 1030 Q2  =  UWIND**2 + TWIND2
      IF ( XMACH .EQ. 0.)  GO TO 1040
      CPP(J) = Q000*((1.+Q001*(1.-Q2))**3.5-1.)
      GO TO 1050
 1040 CPP(J) = 1. - Q2
 1050 CONTINUE
C
C -----------------
      RETURN
C -----------------
      END
C
      SUBROUTINE CVEL(M,N,NTAPEX,A,B,C,CL,U,V,W)
C
C ..................................................................
C     COMPUTES VELOCITY COMPONENTS FOR A GIVEN PANEL PRESSURE DIFFERENCE
C ..................................................................
C
      DIMENSION A(1),B(1),C(1),U(1),V(1),W(1),CL(1)
      DO 1000 J=1,M
      U(J)=0.0
      V(J)=0.0
      W(J)=0.0
 1000 CONTINUE
C
```

```
      DO 1010 J=1,N
      READ (NTAPEX) (A(I),B(I),C(I),I=1,N)
      DO 1010 I=1,N
      U(I)=U(I)+A(I)*CL(J)
      V(I)=V(I)+B(I)*CL(J)
      W(I)=W(I)+C(I)*CL(J)
 1010 CONTINUE
      JC =(J/10)*10 - J
 1020 CONTINUE
C
C ------------------
      RETURN
C ------------------
      END
      SUBROUTINE DCPD(NN,NTAPEX,A,ALPHAN,CLN)
C
C ..................................................................
C     COMPUTES WING PANEL PRESSURE DIFFERENCE
C ..................................................................
C
      DIMENSION A(1),ALPHAN(1),CLN(1)
C
      DO 1000 J=1,NN
      CLN(J)=0.0
 1000 CONTINUE
C
      DO 1010 J=1,NN
      READ (NTAPEX) (A(I),I=1,NN)
      DO 1010 I=1,NN
      CLN(I)=CLN(I)+A(I)*ALPHAN(J)
 1010 CONTINUE
C
C ------------------
      RETURN
C ------------------
      END
C
      SUBROUTINE CANBWD (NB,NW,NTAPEX,NTAPET,A,CLW,ALPHAB,ALPHAW)
C
C ..................................................................
C     FOR WING-BODY CASES, COMPUTES NORMAL VELOCITY COMPONENTS ON WING
C ..................................................................
C
      DIMENSION A(1),ALPHAW(1),ALPHAB(1),CLW(1)
C
      DO 1000 J=1,NW
      ALPHAW(J)=0.
 1000 CONTINUE
      DO 1010 J=1,NW
      READ (NTAPEX) (A(I),I=1,NW)
      DO 1010 I=1,NW
      ALPHAW(I)=ALPHAW(I)+A(I)*CLW(J)
 1010 CONTINUE
      DO 1020 J=1,NB
      READ (NTAPET) (A(I),I=1,NW)
      DO 1020 I=1,NW
      ALPHAW(I)=ALPHAW(I)+A(I)*ALPHAB(J)
 1020 CONTINUE
C
C ------------------
      RETURN
C ------------------
      END
C
      SUBROUTINE DCPI (NB,NW,NTAPEX,A,ALPHAB,CLW,CLB)
```

```
C
C ......................................................................
C     COMPUTES BODY PANEL PRESSURE DIFFERENCE
C ......................................................................
C
      DIMENSION A(1),ALPHAB(1),CLW(1),CLB(1)
C
      DO 1000 J=1,NB
      CLB(J)=0.
 1000 CONTINUE
      DO 1010 J=1,NB
      READ (NTAPEX) (A(I),I=1,NB)
      DO 1010 I=1,NB
      CLB(I)=CLB(I)+A(I)*ALPHAB(J)
 1010 CONTINUE
      DO 1020 J=1,NW
      READ (NTAPEX) (A(I),I=1,NB)
      DO 1020 I=1,NB
      CLB(I)=CLB(I)+A(I)*CLW(J)
 1020 CONTINUE
C
C -----------------
      RETURN
C -----------------
      END
      SUBROUTINE CAMBW(NW,NTAPEX,A,   CLW,ALPHAW)
C
C ......................................................................
C     FOR WING ONLY CASE, COMPUTES NORMAL VELOCITY COMPONENTS ON WING
C ......................................................................
C
      DIMENSION A(1),ALPHAW(1),CLW(1)
C
      DO 1000 J=1,NW
      ALPHAW(J)=0.
 1000 CONTINUE
      DO 1010 J=1,NW
      READ (NTAPEX) (A(I),I=1,NW)
      DO 1010 I=1,NW
      ALPHAW(I)=ALPHAW(I)+A(I)*CLW(J)
 1010 CONTINUE
C
C -----------------
      RETURN
C -----------------
      END
      SUBROUTINE SURINT(  N, XI, YI, CF, NPT, XO, YO, ZO)
C
C          USE THE SURFACE FIT COEFFICIENT CF( XI, YI) FOR THE
C         EVALUATION OF ZO( XO, YO) AT NPT POINTS.
C
      DIMENSION XI(1), YI(1), CF(1), XO(1), YO(1), ZO(1)
C
C
      DO 200 I =1, NPT
      SUM = CF(1) + CF(2)*XO(I) + CF(3)*YO(I)
      DO 100 J =1, N
      J3 = J + 3
      RI = ( XO(I) - XI(J) )**2 + ( YO(I) - YI(J) )**2
      IF ( RI .LT. 1.0E-9 ) GO TO 100
      SUM = SUM + CF(J3) * RI * ALOG( RI )
  100 CONTINUE
      ZO(I) = SUM
  200 CONTINUE
C
```

```
C
      RETURN
      END
      OVERLAY(WANG, 7, 10)
      PROGRAM PBODY
C
C
C        READ INPUT OF FINITE ELEMENT GRID SPECIFICATIONS FROM
C       NSWC PROGRAM * BING * OUTPUT
C
C
      COMMON/ MAIN / NTAPEA,NTAPEB,NTAPEC,NTAPED,NTAPEE,NTAPEF,NTAPEI,
     1         NTAPEO,NBODY,NWING,XMACH,SYM,KACE,NPOLAR,IRW
      COMMON/CORTRN/ NSYM,NTRANS(20),ISYM(20),VA(3,2,20), DTFM(9,2,20)
      COMMON/PBDATA/ NBODYS, XMIDB(51), NTHETB, THETB(15), CPBB(51,15)
     1         ,XC (3,700),NG(350),NELN(700),NE,NGRIDP,PINF,IPUNCH
     2         ,CPB, SID(20)
      COMMON/ELMCNT/ XI(3, 350)
      COMMON/REPEAT/ IREPT
      DIMENSION CPB(150), XS(115,4), YS(115,4), CPBS(115,4),CF(115,4)
     1         , PB(700), DTB(4)
C
C        CHECK SIGN OF SID(NSYM) FOR PRESSURE OUTPUT SIGN AS DESIRED.
C
      ASID = ABS( SID(NSYM) )
      PSIGN = SID(NSYM)/ ASID
      NSID = IFIX( ASID )
      PCONV = 0.5 * 1.4 * PINF * XMACH**2
      IREPT = 1
      IF ( NPOLAR  .GT. 0 ) GO TO 300
C
C        READ IN NUMBER OF GRID ( NGRIDP) AND ELEMENT ( NE ) CARDS.
C
      READ (5,901)  NGRIDP, NE
C
C              CONVERT GRID COORDINATE TO RECTANULAR COORD'NATE XI ( X,
C         Y, Z )
C
      NS = NSYM
      CALL CONVT( XI, NG, NGRIDP, NS,  ISYM)
C
C              TRANSFORM GRID COORDINATE TO THE AERO CODE COORDINATE
C         SYSTEM  XI( X, Y, Z ).
C
      CALL COORDT( XI, NGRIDP )
C
C              COMPUTE THE CENTROID OF ELEMENT XC( X, Y, Z ) .
C
      CALL CENTRO
C
C              EXPRESS CENTROID XC( X, Y, Z ) IN TERMS OF SURFACE
C         PARAMETERS  X  AND  THETA    FOR SURFACE FIT.
C
      CALL CONVBK( XC, NE)
C
C              STORE XMIDB, THETB  AND  CPBB  INTO LOCATIONS OF ONE
C        DIMENSIONAL ARRAY FOR SURFACE FIT.
C
C
C
C              FOR REDUCING COMPUTING TIME OF INVERTING A LARGE MATRIX,
C              THE INTERPOLATION IS DIVIDED INTO SEVERAL PATCHES OF
C              EQUAL OR LESS THAN 110 POINTS.
C
  300 CONTINUE
C
```

```
      NBD = NBODYS*NTHETB
      NCOL = 4
      NSEG = NTHETB / NCOL + 1
      THETB(NTHETB+1) = 360.0
      DO 450 ISEG = 1, NSEG
      NCL1 = 1 + (ISEG-1)*(NCOL-1)
      NCL2 = NCL1 + NCOL - 1
      NBSEG = (NCL2 - NCL1 + 1)*NBODYS
      DTB(ISEG) = THETB(NCL2) - THETB(NCL1)
      DO 400 J = NCL1, NCL2
      DO 400 I = 1, NBODYS
      IJ = I + (J-NCL1)*NBODYS
      XS(IJ, ISEG) = XMIDB(I)
      YS(IJ, ISEG) = (THETB(J) - THETB(NCL1) )/DTB(ISEG)
      CPBS(IJ, ISEG) = CPBB(I, J)
  400 CONTINUE
      CALL SURFIT( XS(1, ISEG), YS(1, ISEG), NBSEG, CPBS(1, ISEG),
     1             CF(1, ISEG) )
  450 CONTINUE
C
      DO 600 I = 1, NE
C
C          SHIFT THE VALUES OF THETB TO 0 - 180 DEGREES
C
      XC2I = XC(2,I)
      IF ( XC(2, I) .GT. 180.0 ) XC2I = 360.0 - XC(2, I)
C          CHECK FOR PROPER SEGMENT
C
      DO 510 K=1, NSEG
      KEND = K*(NCOL-1) + 1
      IF ( XC2I  .LT. THETB(KEND) ) GO TO 520
  510 CONTINUE
  520 CONTINUE
      ISEG = K
C
      XC2I = (XC2I - THETB(KEND-3) )/DTB(ISEG)
      SUM = CF(1,ISEG) + CF(2,ISEG)*XC(1,I) + CF(3,ISEG)*XC2I
      DO 530 L =1, NBSEG
      L3 = L + 3
      RI = ( XC(1,I)-XS(L,ISEG) )**2 + ( XC2I-YS(L,ISEG) )**2
      IF ( RI .LT. 1.0E-10 ) GO TO 530
      SUM = SUM + CF(L3, ISEG) * RI * ALOG( RI )
  530 CONTINUE
      PB(I) = SUM * PCONV * PSIGN
C
C               PUNCH PRESSURE AND GRID POINT IDENTIFICATION IN THE FORM
C          OF BULK DATA FORMAT OF NASTRAN.
C
      IF ( IPUNCH  .EQ. 0 ) GO TO 600
      PUNCH  902, (NSID, PB(I), NELM(I)  )
  600 CONTINUE
C
      WRITE (6,903) ( NELM(I),   XC(1,I),   XC(2,I), PB(I), I=1,NE)
  902 FORMAT (8HPLOAD2  ,I8,F8.2,I8)
  903 FORMAT (1H1//'      AERODYNAMIC PRESSURE ON THE BODY - - CENTROID '
     1         ,'OF FINITE ELEMENT AS GENERATED BY NSWC CODE IN THE'/
     A       7X,'SAME LOCAL COORDINATE SYSTEM'///3X,
     2         'ELEMENT',6X,'X',12X,'THETA',9X,'PRESSURE',13X,'ELEMENT',
     3         6X,'X',12X,'THETA',9X,'PRESSURE'//(1H0,2(I8,E12.3,2X,
     4         E12.3,4X,E12.3,10X) ) )
  901 FORMAT (2I10)
C
C
      END
C
```

```
      SUBROUTINE CONVRK( X, NE)
C
C
C           CONVERT COORDINATE X, Y, Z INTO X AND THETA FOR SURFACE
C         FIT PURPOSES.
C
      DIMENSION X( 3, NE)
C
      RD = 57.2957795
C
      DO 100 I = 1, NE
      R = SQRT( X(2, I) **2 + X( 3, I)**2 )
      ANG = ACOS ( X( 3, I) / R ) * RD
      IF ( X( 2, I)  .GT.  0.0 ) X( 2, I) = ANG
      IF ( X( 2, I)  .LT.  0.0 ) X( 2, I) = 360.0 - ANG
  100 CONTINUE
C
C
      RETURN
      END
C
      OVERLAY(WANG, 7, 11)
      PROGRAM PSUM
C
C
C          THIS SUBROUTINE INTERPOLATES THE PRESSURE COEFFICIENTS CPB (
C          EFFECT OF WING ON THE BODY ) AT THE BODY STATIONS AND ADDS TO
C          THAT OF THE BODY ALONE CPBB.
C
      COMMON/ MAIN / LA, LB, LC, LD, LE, LF, LI,LO, NBODY, NWING, XMACH,
     1          SYM, KACE, NPOLAR, IRW
      COMMON/ NVAR / NFMT(9), NROW(2), NS, NPANEL, NACEL, NROWB, NROWW,
     1          NCOLW, NTHETA, NTHETS, NXLE, NRG, NPOLR, NCLX
      COMMON /TVAR/  TITLE(20),THETA(210),THETAB(15),THETAS(15),THETAN(
     1              11,2),TCL(11),TCD(11),TCM(11), THICK, TWIST
      COMMON /XVAR/  XBAR(210),XC(210),XMI(2),XMIN(2),XMTN(2),XB(51),XN(
     1               51,2),XNNCD(2),XNNCL(2),XNNCM(2),XYZ(3),XCL(11),XCD
     2               (11), XNACEL,XP,XCPBAR
      COMMON/PBDATA/ NBODYS, XMIDB(51), NTHETB, THETB(15), CPBB(51,15)
     1,XCB(3,700),NG(350),NELM(700),NE,NGRIDP,PINF,IPUNCH, CPB
      COMMON/REPEAT/ IREPT
      DIMENSION XADD(115), YADD(115), CF(115), CPB(150), FN(115,2)
C
C
      IREPT = 1
      CDR = 57.2957795
      NTHETA = NBODY / NROWB
C
C         CHECK THE TOTAL NUMBER OF POINT OF CPB AND DIVIDE INTO 2
C       SEGMENTS IF IT EXCEEDS 110
C
      LOOP = 0
      NSTART = 1
      NSEG = 1
      NCL1 = 1
      NCL2 = NTHETA
      NCOL1 = NCL2 - NCL1 + 1
      NCL21 = 1
      NCL22 = NTHETB
      NCOL2 = NCL22 - NCL21 + 1
      NBD = NBODY
      IF ( NBODY .LT. 110 ) GO TO 70
      NSEG = NBODY/110 + 1
      NCOL1 = NTHETA/NSEG + 1
      NCOL2 = NTHETB/NSEG
```

```
C
C        SCALE XC AND THETAB FOR BETTER SURFACE FIT
C
   50 CONTINUE
      NCL1 = 1
      NCL2 = NCOL1 + NCL1 - 1
      NCL21 = 1
      NCL22 = NCOL2 + NCL21 - 1
      NBD = NCOL1*NROWB
      GO TO 70
C
   60 CONTINUE
      NCL1 = NCL2 - 1
      NCL2 = NTHETA
      NCOL1 = NCL2 - NCL1 + 1
      NCL21 = NCL22 + 1
      NCL22 = NTHETB
      NCOL2 = NCL22 - NCL21 + 1
      NBD = NCOL1*NROWB
      NSTART = NROWB*(NCL1-1) + 1
C
   70 CONTINUE
      CALL SCALE(NROWB, XC, FN(1,1) )
      DTB = THETB(NCL22) - THETB(NCL21)
      JJ = 1
      DO 80 J=NCL1, NCL2
      FN(JJ, 2) = ( THETAB(J) - THETB(NCL21) )/DTB
      JJ = JJ + 1
   80 CONTINUE
C
      DO 100 I=1,NCOL1
      DO 100 J=1,NROWB
      IJ = J + (I-1)* NROWB
      XADD(IJ) = FN(J,1)
      YADD(IJ) = FN(I,2)
  100 CONTINUE
C
C
C
      CALL SURFIT( XADD, YADD, NBD, CPB(NSTART), CF)
C
C         SCALE XMIDB AND THETB FOR CORRECT INTERPOLATION
C
      IC = 1
      DXC = XC(NROWB) - XC(1)
      DO 105 I=1, NBODYS
      IF ( XMIDB(I) .LT. XC(1) .OR. XMIDB(I) .GT. XC(NROWB) ) GO TO
     1         105
      IF ( IC .EQ. 1 ) NBS = I - 1
      FN(IC,1) = (XMIDB(I) - XC(1) )/DXC
      IC = IC + 1
  105 CONTINUE
      NXT = IC - 1
      DTB = THETAB(NTHETA) - THETAB(1)
      DO 110 I=NCL21,NCL22
      FN(I,2) = (THETB(I) - THETB(NCL21) )/DTB
  110 CONTINUE
      DO 120 J =NCL21, NCL22
      DO 120 I =1,NXT
      SUMB = CF(1) + CF(2) * FN(I,1) + CF(3) * FN(J,2)
      DO 130 K=1,NBD
      K3 = K+3
      RI = ( FN(I,1) - XADD(K) )**2 + ( FN(J,2) - YADD(K) )**2
      IF ( RI .LT. 1.E-5 ) GO TO 130
      SUMB = SUMB + CF(K3) * RI * ALOG( RI )
  130 CONTINUE
      II = I + NBS
      CPBB(II,J) = SUMB + CPBB(II,J)
  120 CONTINUE
      LOOP = LOOP + 1
      IF ( NSEG .GT. 1 .AND. LOOP .EQ. 1 ) GO TO 60
C
      WRITE (6,1830)
      WRITE (6,1860) ( THETB(I), I=1,NTHETB)
      WRITE (6,1870)
      NBS = NBODYS
      IF ( XMACH  .LT. 1.0 ) NBS = NBODYS - 1
      DO 200 I=1,NBS
      WRITE (6,1880) XMIDB(I), ( CPBB(I,J), J=1,NTHETB)
  200 CONTINUE
C
C
 1830 FORMAT (1H0/2X,*BODY PRESSURE COEFFICIENTS (CP), INCLUDING THE EF*
     1          ,*FECT OF WING*)
C
 1860 FORMAT (1H0,4X,*THETA (DEG)*,F14.4,9F10.4/(F29.4,9F10.4) )
C
 1870 FORMAT (1H ,9X,*X*)
C
 1880 FORMAT (1H ,F14.4,F15.5,9F10.5/(F29.5,9F10.5) )
C
      END
      SUBROUTINE SCALE( NX, FN1, FN2)
C
C        GENERATES FN2 FROM FN1 BY SCALING TO A RANGE BETWEEN 0 AND 1
C
C
      DIMENSION FN1(1), FN2(1)
C
      FMAX = FN1(1)
      FMIN = FN1(1)
      DO 100 I=2, NX
      FMIN = AMIN1( FMIN, FN1(I) )
      FMAX = AMAX1( FMAX, FN1(I) )
  100 CONTINUE
C
      DFN = FMAX - FMIN
      DO 200 I=1, NX
      FN2(I) = ( FN1(I) - FMIN )/DFN
  200 CONTINUE
C
C
      RETURN
      END
C
      OVERLAY(WANG, 7, 12)
      PROGRAM WINGPR
C
C
C        READ IN FINITE ELEMENT , INTERPOLATE PRESSURE AND GENERATES
C        PRESSURE LOAD ON THE WING.
C
      COMMON/ MAIN / LA, LB, LC, LD, LE, LF, LI,LO, NBODY, NWING, XMACH,
     1          SYM, KACE, NPOLAR, IRW
      COMMON/CORTRN/ NSYM,NTRANS(20),ISYM(20),VA(3,2,20), BTFM(9,2,20)
      COMMON/NVAR / NFMT(9), NROW(2), NS, NPANEL, NACEL, NROWB, NROWW,
     1          NCOLW, NTHETA, NTHETS, NXLE, NRG, NPOLR, NCLX
      COMMON/PBDATA/ NBODYS,XMIDB(51), NTHETB,THETB(15),CPBB(51,15)
     1          , XCB(3,700), NG(350), NELM(700), NE,NGRIDP,PINF,IPUNCH
     2          , CPB
```

```
      COMMON/ELMCNT/ XCI(3, 350)
      COMMON/WINGOT/ CPU, CPL,NWG,ISOLID,IFORM(10),YAW,BROLL
      COMMON /XVAR/  XBAR(210),XC(210),XNI(2),XNXN(2),XNTN(2),XB(51),XN(
     1               51,2),XNNCD(2),XNNCL(2),XNNCM(2),XYZ(3),XCL(11),XCD
     2               (11), XNACEL,XP,XCPBAR
      COMMON/YVAR/ YBAR(210), YC(210), YNI(2)
      COMMON/ZVAR/ ZBAR(210), ZC(210),ZNI(2),ZDELTA(51),ZDN(51,2),ZP,ZA
      DIMENSION CPU(110), CPL( 110)
      DIMENSION CPB(150)
C
      CALL WINGO( XC, YC)
C
      END
C
      SUBROUTINE WINGO( XC, YC)
C
C         INTERPOLATE OUTPUT FOR NASTRAN USE, GRID POINTS AND ELEMENT
C         LOCATIONS ARE GENERATED FROM PING.
C
      COMMON/ MAIN / LA, LB, LC, LD, LE, LF, LI,LO, NBODY, NWING, XMACH,
     1          SYM, KACE, NPOLAR, IRW
      COMMON/COEFFB/ CF(100, 4, 2)
      COMMON/CORTRN/ NSYM,NTRANS(20),ISYM(20),VA(3,2,20), DTFM(9,2,20)
      COMMON/NVAR / NFMT(9), NROW(2), NS, NPANEL, NACEL, NROWB, NROWW,
     1          NCOLW, NTHETA, NTHETS, NXLE, NRG, NPOLR, NCLX
      COMMON/PCOORD/ X(210), Y(210), NPT
      COMMON/WINGOT/ CPU, CPL,NWG,ISOLID,IFORM(10),YAW,BROLL
      COMMON/PBDATA/ NBODYS,XMIDB(51), NTHETB,THETB(15),CPBB(51,15)
     1          , XCB(3,700), NG(350), NELM(700), NE,NGRIDP,PINF,IPUNCH
     2          , CPB
      COMMON/REPEAT/ IREPT
      DIMENSION CPB(150)
      DIMENSION CPU(110), CPL(110), XC(1), YC(1)
      DIMENSION NELMT( 6)
      DIMENSION COF(115)
C
      PCONV = 0.5 * 1.4 * PINF * XMACH * XMACH
      NADJ = 0
      IF ( XMACH .GT. 2.5 ) NADJ = NBODY
      IREPT = 0
C
C
      NW = 1
      NPT = ( NPANEL - NBODY ) / NWG
   10 CONTINUE
C
C         STORE  PRESSURE AT CONTROL POINT AND ITS LOCATION FOR
C         INTERPOLATION.
C
      NC1 = NPT
      NC3 = NC1 + 3
C
      IF ( ISOLID  .EQ. 0 ) GO TO 205
C
C        SOLID WING CASE , LIFTING PRESSURE
C
      DO 200 J=1,NC1
      JA = J + ( NW-1)*NPT
      COF (J) = ( CPL(JA) - CPU(JA) ) * PCONV
      JB = JA + NBODY - NADJ
      X(J) = XC(JB)
      Y(J) = YC(JB)
  200 CONTINUE
C
      WRITE (6,803) ( X(I), Y(I), COF(I), I=1,NC1)
```

```
C
      CALL SURFIT( X, Y, NC1, COF, CF(1, NW, 1) )
C
      GO TO 250
C
  205 CONTINUE
      DO 210  J =1, NC1
      JA = J + ( NW-1)*NPT
      COF(J) = CPU(JA) * PCONV
      JB = JA + NBODY - NADJ
      X(J) = XC(JB)
      Y(J) = YC(JB)
  210 CONTINUE
C
      WRITE (6,801)  ( X(I), Y(I), COF(I), I=1, NC1 )
C
      CALL SURFIT( X, Y, NC1, COF, CF(1, NW, 1) )
C
C
      DO 215  J =1, NC1
      JA = J + ( NW-1)*NPT
      COF(J) = CPL(JA) * PCONV
  215 CONTINUE
C
      WRITE (6,802)  ( X(I), Y(I), COF(I), I=1, NC1 )
C
      CALL SURFIT( X, Y, NC1, COF, CF(1, NW, 2) )
C
C
  250 CONTINUE
C
C         COMPUTES PRESSURE AT DESIRED LOCATION ( CENTROID OF ELEMENT ),
C         FOR EACH COMPLETE WING.
C
      CALL PWING( NW )
C
      NW = NW + 1
C
C
      IF ( NW  .LE.  NWG ) GO TO 10
C
  801 FORMAT (1H0,* PRESSURE AT CONTROL POINT - UPPER SURFACE*/ 15X,
     1         *X*,9X,*Y*,9X,*PRESSURE*///(11X,F8.4,2X,F8.4,2X,E12.4) )
  802 FORMAT (1H0,* PRESSURE AT CONTROL POINT - LOWER SURFACE*/ 15X,
     1         *X*,9X,*Y*,9X,*PRESSURE*///(11X,F8.4,2X,F8.4,2X,E12.6) )
  803 FORMAT (1H1,*     TOTAL LIFTING PRESSURE ON THE WING AT THE*
     A         * CONTROL POINT*//15X,*X*,
     1         9X,*Y*,9X,*PRESSURE*///(11X,78.4,2X,F8.4,2X,E12.4) )
C
      RETURN
      END
C
      SUBROUTINE PWING( NW )
C
C         INTERPOLATES PRESSURE ON WING AT GIVEN LOCATION X,Y FOR
C         LOCATION XC, YC OR CENTROID OF ELEMENT
C
      COMMON/COEFFB/ CF(100, 4, 2)
      COMMON/CORTRN/ NSYM,NTRANS(20),ISYM(20),VA(3,2,20), BTFM(9,2,20)
      COMMON/PBDATA/ NBODYS, XNIDB'51), NTHETB, THETB(15), CPBB(51,15)
     1,XCB(3,700),NG(350),NELM(700),NE,NGRIDP,PINF,IPUNCH, CPB
     2          , SID(20)
      COMMON/ELMCNT/ XI(3, 350)
      COMMON/PCOORD/ X(210), Y(210), NP
      COMMON/WINGOT/ CPU, CPL,NWG,ISOLID,IFORM(10),YAW,BROLL
```

AD-A081 596 MCDONNELL DOUGLAS ASTRONAUTICS CO HUNTINGTON BEACH CA F/6 9/2
AERODYNAMIC COMPUTER CODE FOR COMPUTING PRESSURE LOADS ON A MIS--ETC(U)
DEC 79 K K WANG N00014-75-C-0462
UNCLASSIFIED MDC-G8223 NL

3 of 3
AD
A081596

END
DATE
FILMED
4 80
DTIC

```
      DIMENSION CPU(110), CPL(110), CPB(150), P(210)
      INTEGER CHECK, WORD1
C
      DATA  WORD1/ 5HCQUAD/
C
C        WRITE TITLE OF WING PRESSURE OUTPUT
C
      IF ( NW  .GT. 1 ) GO TO 50
      WRITE (6,901)
      WRITE (6,908)  NWG
   50 CONTINUE
      WRITE (6,910)  NW
      NPUNCH = 0
      NSID = IFIX( SID(NSYM) )
      NS = 2
      IF ( ISOLID  .NE. 0 ) NS = 1
      DO 150 K=1,NS
C
C        CHECK SIGN OF SID(NSYM) FOR PRESSURE OUTPUT SIGN AS DESIRED.
C
      ASID = ABS( SID(NSYM) )
      PSIGN = SID(NSYM)/ ASID
      NSID = IFIX( ASID )
C
C
C              READ IN FINITE ELEMENT SPECIFICATIONS FROM PING OUTPUT
C
C
C         READ IN NUMBER OF GRID (NG) AND ELEMENT SPECIFICATION (NE)
C         CARDS.
C
      READ (5,902)  NGRIDP,  NE
C
      CALL CONVT( XI, NG, NGRIDP, NSYM, ISYM)
C
      CALL COORDT( XI, NGRIDP)
C
      CALL CENTRO
C
      DO 100 IE = 1,NE
C
C        CHANGE THE SIGN OF Y COORDINATE OF THE LEFT WING TO REFLECT
C        THE SYMMETRY OF WING ARRANGEMENT
C
      YCOORD = ABS( XCB( 2, IE ) )
C
      SUM = CF(1,NW,K) + CF(2,NW,K)*XCB(1,IE) + CF(3,NW,K)*YCOORD
      DO 110 J = 1, NP
      J3 = J + 3
      RI = ( XCB(1,IE) - X(J) )**2 + ( YCOORD - Y(J) )**2
      IF ( RI  .LT. 1.E-9 ) GO TO 110
      SUM = SUM + CF( J3, NW, K )* RI * ALOG( RI )
  110 CONTINUE
      P(IE) = SUM * PSIGN
  100 CONTINUE
      NSYM = NSYM + 1
C
      IF ( ISOLID  .EQ. 0  .AND.  K  .EQ. 1 ) WRITE (6,903)
      IF ( ISOLID  .EQ. 0  .AND.  K  .EQ. 2 ) WRITE (6,904)
C
      WRITE (6,909)
      WRITE (6,906) ( NELM(J), XCB(1,J), XCB(2,J), P(J), J=1,NE )
C
C         PUNCH  PRESSURE AND GRID POINT IDENTIFICAITON IN THE FORM OF
C         BULK DATA FORMAT OF NASTRAN
```

```
C
      IF ( IPUNCH  .EQ.  0 ) GO TO 150
      DO 220  L=1, NE
      PUNCH 807, ( NSID, P(L), NELM(L) )
  220 CONTINUE
      NPUNCH = NPUNCH + NE
  150 CONTINUE
C         WRITE NUMBER OF BULK DATA CARDS PUNCHED
C
      IF ( IPUNCH  .EQ.  1 ) WRITE (6,905) NPUNCH
C
C
C
C
  807 FORMAT ( 8HPLOAD2  ,I8,F8.2,I8 )
  901 FORMAT (1H1//*      AERODYNAMIC PRESSURE ON THE WING - - CENTROID *
     1              ,*OF FINITE ELEMENT AS GENERATED BY NSWC CODE IN THE*/
     A          7X,*SAME LOCAL COORDINATE SYSTEM*/)
  902 FORMAT  (2I10 )
  903 FORMAT (1H0,10X,*PRESSURE LOAD ON THE UPPER SURFACE OF WING*//)
  904 FORMAT (///1H0,10X,*PRESSURE LOAD ON THE LOWER SURFACE OF WING*//)
  906 FORMAT (1H0,2(I7,E13.3,3X,E13.3,4X,E13.3,6X) /(1X,2(I7,E13.3,3X,
     1              E13.3,4X,E13.3,6X)   ) )
  907 FORMAT (1H0,10X,*SOLID WING, SUM OF UPPER AND LOWER SURFACE*,
     1              * PRESSURES*//)
  908 FORMAT (1H0,10X,*TOTAL NUMBER OF WINGS  *,I5/)
  909 FORMAT (1H0/3X,*ELEMENT*,9X,*X*,11X,*Y*,11X,*PRESSURE*,13X,
     1              *ELEMENT*,9X,*X*,11X,*Y*,11X,*PRESSURE*//)
  910 FORMAT (1H0,10X,*WING  NO. *,I5/ )
C
  905 FORMAT (1H ///*          NUMBER  OF  CARD  PUNCHED  FOR  THIS  RUN
     1              *, I10 /)
C
      RETURN
      END
C
      OVERLAY(WANG, 7, 13)
      PROGRAM SAVE
C
C
      COMMON/ MAIN / NTAPEA,NTAPEB,NTAPEC,NTAPED,NTAPEE,NTAPEF,NTAPEI,
     1          NTAPEO,NBODY,NWING,XMACH,SYM,KACE,NPOLAR,IRW
      COMMON / BODYSP/ DRDX(51), DZDX(51), DRDXC(51)
      COMMON /AVAR/  A(210),ACB(21),ABX(100),AWX(110),AREA(210),ARN(2),A
     1               RWT(20),ALPHA(210),ALPHAB(210),ALPHAT(110),
     2               ALPHAW(110),ALPHAX(110),AWS(110),ABT(100),ANTH(
     3               210,2),ALPHAA,ALPHAD,ARA,ARADEG,ARB,ARW(10), ARAS,
     4               AT, AAA
      COMMON /BODS/ NUMBOD,XNOS(3),YNOS(3),ZNOS(3),NBSTAT(3)
      COMMON /BVAR/  B(210), BBCL, BBCD, BBCY, BBCM, BBCMR, BBCMY
      COMMON /CVAR/  C(210),CHORD(210),CL(210),CPNN(1,1,1),CLS(210)
     1               ,CDR,CASE,CPCALC,CBAR,CONSNT,CLBAR,CLX,CLM,TDM
      COMMON /DVAR/  D(210),DZDXB(51), DADEG, DARAD
      COMMON /IVAR/  IPOLAR
      COMMON /KVAR/  KASE,KONFIG,KPOLAR
      COMMON /NVAR/  NFMT(9),NROW(2)
     1               ,NS,NPANEL,NACEL,NROWB,NROWW,NCOLW,NTHETA,NTHETS,
     2                     NXLE,NRG,NPOLR ,NCLX
      COMMON /PVAR/  POLAR
      COMMON /RVAR/  R(51),RN(51,2), RATIOX, RFAREA
      COMMON /SVAR/  SEMIS,SLC
      COMMON /TVAR/  TITLE(20),THETA(210),THETAB(15),THETAS(15),THETAN(
     1             11,2),TCL(11),TCD(11),TCM(11),THICK,TWIST
      COMMON /UVAR/  UBWT(150),UWBT(110),UWWT(110),UNCL(1,1)
      COMMON /VVAR/  VBWT(150),VWBT(110),VWWT(110),VNCL(1,1)
     1                 ,VOUT
      COMMON /WVAR/  WBWT(150),WWBT(110),WWWT(110),WNCL(1,1)
      COMMON /XVAR/  XBAR(210),XC(210),XNI(2),XNXN(2),XNTN(2),XB(51),XN(
     1                51,2),XNNCD(2),XNNCL(2),XNNCM(2),XYZ(3),XCL(11),XCD
     2                (11), XNACEL, XP, XCPBAR
      COMMON /YVAR/  YBAR(210),YC(210),YNI(2)
      COMMON /ZVAR/  ZBAR(210),ZC(210),ZNI(2),ZDELTA(51),ZDN(51,2)
     1                ,ZP,ZA
      COMMON/PBDATA/ NBODYS, XMIDB(51), NTHETB, THETB(15), CPBB(51,15)
     1,XCB(3,700),NG(350),NELM(700),NE,NGRIDF,PINF,IPUNCH, CPB
      COMMON/CORTRN/ NSYM,NTRANS(20),ISYM(20),VA(3,2,20), BTFM(9,2,20)
      COMMON/WINGOT/ CPU, CPL,NWG,ISOLID,IFORM(10),YAW,BROLL
      COMMON/EXTRA/ ARAN, DSB1, DSB2, DS1, DS2, NB1,NB2,NS1,NS2,
     ,          NTN, NXN, XNOSE
C
      DIMENSION CPU(110), CPL( 110),CPB(150),XNS1(2),XNS2(2),NS1(2),
     1          NS2(2), DS1(2), DS2(2)
C
      REWIND 12
      IF ( IRW  .EQ. 2 ) GO TO 100
C
      WRITE(12) DATE,NTAPEA,NTAPEB,NTAPEC,NTAPED,NTAPEE,NTAPEF,NTAPEI,
     1     NTAPEO,NBODY,NWING,XMACH, SYM,KACE, DRDX, DZDX, NPOLAR, A,
     2          ACB, ABX, AWX, AREA, ARN,ARWT, ALPHA, ALPHAB, ALPHAS,
     3          ALPHAT,ALPHAW, ALPHAX, AWS,ABT,AN1N,ALPHAA,ALPHAD,ARA,
     4          ARADEG,ARB,ARW,ARAS,AT,AAA,NUMBOD,XNOS,YNOS,ZNOS,NBSTAT,
     5    B,BBCL,BBCD,BBCM,C,CHORD,CL,CPNN,CLS,CDR,CASE,CPCALC,CBAR,
     6    CONSNT,CLBAR,CLX,CLM,CDM,D,DZDXB,DADEG,DARAD,IPOLAR,KASE,KONFIG
     7    ,KPOLAR,NFMT,NROW,NS,NPANEL,NACEL,NROWB,NROWW,NCOLW,NTHETA,
     8    NTHETS,NXLE,NRE,NPOLR,NCLX,POLAR,R,RN,RATIOX,RFAREA,SEMIS,SLC,
     9    TITLE,THETA,THETAB,THETAS,THETAN,TCL,TCD,TCM,THICK,TWIST,UBWT,
     A    UWBT,UWWT,UNCL,VBWT,VWBT,VWWT,VNCL,VOUT,WBWT,WWBT,WWWT,WNCL,
     B    XBAR,XC,XNI,XNXN,XNTN,XB,XN,XNNCD,XNNCL,XNNCM,XYZ,XCL,XCD,XP,
     C    XNACEL,XCPBAR,YBAR,YC,YNI,ZBAR,ZC,ZNI,ZDELTA,ZDN,ZP,ZA,NBODYS,
     D    XMIDB,NTHETB,THETB,CPBB,XCB,NG,NELM,NE,NGRIDP,PINF,IPUNCH,CPB,
     E    NSYM,NTRANS,ISYM,VA,BTFM,CPU,CPL,NWG,ISOLID,ARAN,DSB1,DSB2,DS1,
     F    DS2,NB1,NB2,NS1,NS2,NTN,NXN,XNOSE
C
      GO TO 200
C
  100 CONTINUE
C
      READ (12) DATE,NTAPEA,NTAPEB,NTAPEC,NTAPED,NTAPEE,NTAPEF,NTAPEI,
     1     NTAPEO,NBODY,NWING,XMACH, SYM,KACE, DRDX, DZDX, NPOLAR, A,
     2          ACB, ABX, AWX, AREA, ARN,ARWT, ALPHA, ALPHAB, ALPHAS,
     3          ALPHAT,ALPHAW, ALPHAX, AWS,ABT,AN1N,ALPHAA,ALPHAD,ARA,
     4          ARADEG,ARB,ARW,ARAS,AT,AAA,NUMBOD,XNOS,YNOS,ZNOS,NBSTAT,
     5    B,BBCL,BBCD,BBCM,C,CHORD,CL,CPNN,CLS,CDR,CASE,CPCALC,CBAR,
     6    CONSNT,CLBAR,CLX,CLM,CDM,D,DZDXB,DADEG,DARAD,IPOLAR,KASE,KONFIG
     7    ,KPOLAR,NFMT,NROW,NS,NPANEL,NACEL,NROWB,NROWW,NCOLW,NTHETA,
     8    NTHETS,NXLE,NRG,NPOLR,NCLX,POLAR,R,RN,RATIOX,RFAREA,SEMIS,SLC,
     9    TITLE,THETA,THETAB,THETAS,THETAN,TCL,TCD,TCM,THICK,TWIST,UBWT,
     A    UWBT,UWWT,UNCL,VBWT,VWBT,VWWT,VNCL,VOUT,WBWT,WVBT,WWWT,WNCL,
     B    XBAR,XC,XNI,XNXN,XNTN,XB,XN,XNNCD,XNNCL,XNNCM,XYZ,XCL,XC9,XP,
     C    XNACEL,XCPBAR,YBAR,YC,YNI,ZBAR,ZC,ZNI,ZDELTA,ZDN,ZP,ZA,NBODYS,
     D    XMIDB,NTHETB,THETB,CPBB,TCB,NG,NELM,NE,NGRIDP,PINF,IPUNCH,CPB,
     E    NSYM,NTRANS,ISYM,VA,BTFM,CPU,CPL,NWG,ISOLID,ARAN,DSB1,DSB2,DS1,
C
     F    DS2,NB1,NB2,NS1,NS2,NTN,NXN,XNOSE
C
  200 CONTINUE
      REWIND 12
C
      END
      OVERLAY(WANG, 7, 14)
      PROGRAM WRITTP
C
C
C         WRITE ON TAPE FOR GRAPHIC REPRESENTATION BY 3 D PLOTTING
C
      COMMON/ MAIN / LA, LB, LC, LD, LE, LF, LI,LO, NBODY, NWING,
     1          XMACH, SYM, KACE, NPOLAR, IRW, IPLOT
      COMMON /TVAR/  TITLE(20),THETA(210),THETAD(15),THETAS(15),THETAN(
     1              11,2),TCL(11),TCD(11),TCM(11),THICK,TWIST
      COMMON /XVAR/  XBAR(210),XC(210),XNI(2),XNXN(2),XNTN(2),XB(51),XN(
     1               51,2),XNNCD(2),XNNCL(2),XNNCM(2),XYZ(3),XCL(11),XCD
     2               (11), XNACEL, XP, XCPBAR
      COMMON /YVAR/  YBAR(210),YC(210),YNI(2)
      COMMON/PBDATA/ NBODYS,XMIDB(51), NTHETB,THETB(15),CPBB(51,15)
     1         ,XCB(3,700),NG(350),NELM(700),NE,NGRIDP,PINF,IPUNCH,CPB
     2         , SID(20)
      COMMON/WINGOT/ CPU, CPL,NWG,ISOLID,IFORM(10),YAW,BROLL
      COMMON/REPEAT/ IREPT
      COMMON/ NVAR / NFMT(9), NROW(2), NS, NPANEL, NACEL, NROWB, NROWW,
     1          NCOLW, NTHETA, NTHETS, NXLE, NRG, NPOLR, NCLX
C
      DIMENSION CPB(150), CPU(110), CPL(110)
C
      NTB = NTHETB
      NTA = NTHETA
      IF ( YAW .GT. 0.01 ) NTB = 2*(NTHETB-1)
      IF ( YAW .GT. 0.01 .OR. BROLL .GT. 0.01) NTA = 2*NTHETA
      IF ( IREPT .EQ. 2 ) GO TO 200
      REWIND 12
C
C         PRESSURE ON THE WING SURFACE CPU, CPL
C
      NPWING = (NPANEL - NBODY)/NWG
      DO 100 K=1, NWG
      JA = (K-1)*NPWING
      JB = JA + NBODY
      WRITE (12) ( XC(I+JB), YC(I+JB), CPU(I+JA), I=1,NPWING)
      WRITE (12) ( XC(I+JB), YC(I+JB), CPL(I+JA), I=1,NPWING)
  100 CONTINUE
C
C         PRESSURE ON THE BODY, BODY ALONE
C
  200 CONTINUE
C
      WRITE (12) ( (XMIDB(I), THETB(J), CPBB(I,J), J=1,NTB), I=1,
     1                NBODYS)
C
      IF ( IREPT .EQ. 2 ) GO TO 300
C
C         PRESSURE ON THE BODY, DUE TO THE PRESENCE OF WING
C
      DO 250 I=1, NTA
      IJ = (I-1)*NROWB
      WRITE (12) ( XC(J), THETAD(I), CPB(IJ+J), J=1, NROWB)
  250 CONTINUE
C
  300 CONTINUE

C
      END
      OVERLAY(WANG, 7, 15)
      PROGRAM HYSUP
C
C         HIGH SUPERSONIC FLOW ANALYSIS FOR MACH NO. 2.0 - 6.0
C
```

```
      COMMON/ MAIN / LA,LB,LC,LD,LE,LF,LI,LO,NBODY, NWING,XMACHN,
     1          SYMM, KACE, NPOLAR, IRW
      COMMON/CONST /DR, RD, G, GP1, GM1, GR, AV1, AV2, AV3
      COMMON/FREES /XMACH, PF, ARBYAW, BETA, NEND
      COMMON/FLOW  /FLOWFD(10, 2), X(51), Y(51)
      COMMON / BODYSP/ DRDX(51), DZDX(51), DRDXC(51),XWROOT.XWTIP
      COMMON /NVAR/  NDUM(12), NPANEL, NACEL, NROWB, NC, NB
      COMMON/PBDATA/NBODYS, XMIDB(51), NTHETB, THETB(15), CPBB(51,15)
     1         ,XCB(3,700),NG(350),NELM(700),NE,NGRIDP,PINF,IPUNCH
     2        ,CPB, SID(20)
      COMMON/XVAR  /DUMMY(210), XC(210), DUM1(6), XB(51)
      COMMON/YVAR  /DUM2(210), YC(210)
      COMMON/WINGOT/ CPW,NWG,ISOLID,IFORM(10),YAW,BROLL,SLOPE(51)
C
      DIMENSION CPW(110,2), ALPHA(110,2), ALPHAT(110), ARWT(110),
     1           AREA(210), XCR(40), CPB(150)
C
      DATA DR, RD/0.01745329252,  57.29577957/
      DATA AV1, AV2, AV3 / 225.0, 8.0382436E-10, 1.0/
C
      G = 1.4
      GP1 = G + 1.0
      GM1 = G - 1.0
      GR = SQRT( GP1/GM1 )
C
C
      READ (11,800) XMACH, PINF, POLAR, PUNCH, THICK, CAMB, TWIST,
     1               SOLID, WING, XNC, XNB, BODYS, BTHET,(SID(K), K=1,
     2               20), (IFORM(K), K=1,10), XMOM, ARB, YAW, BROLL
C
      XMACHN = XMACH
      BT2 = 0.5*G*XMACH**2
      NBODY = IFIX(BODYS)
      NPOLAR = IFIX(POLAR)
      IPUNCH = IFIX(PUNCH)
      ITHICK = IFIX(THICK)
      ICAMBR = IFIX(CAMB)
      ITWIST = IFIX(TWIST)
      ISOLID = IFIX(SOLID)
      NWING = IFIX(WING)
      NC = IFIX(XNC)
      NB = IFIX(XNB)
      NBODYS = IFIX(BODYS)
      NTHETB = IFIX(BTHET)
      TANA = TAN( ARB*DR )
      TANY = TAN( YAW*DR )
      A1 = 1.0/SQRT( 1.0 + TANA**2 + TANY**2 )
      ARBYAW = ACOS(A1)*RD
C
      READ (11,800) (X(K), K=1,NBODYS)
      READ (11,800) (Y(K), K=1,NBODYS)
      READ (11,800) (THETB(K), K=1,NTHETB)
C
      X(1) = Y(1) = 0.0
      YP = Y(1)
      DO 50 I=2, NBODYS
      XMIDB(I) = 0.5*( X(I) + X(I-1) )
      YN = Y(I)
      Y(I) = 0.5*( YP + YN )
      YP = YN
   50 CONTINUE
      XMIDB(1) = 0.0
      CALL SECEXP(2)
C
      NW2 = 2*NWING
      NC1 = NC - 1
      NB1 = NB - 1
      NBC = NB1*NC1
      NPANEL = NBODY + NBC
      RFAREA = 0.0
      CBAR = 1.0
      DO 200 I = 1, NW2, 2
C
      READ (11,800) (XC(K), K=1,NBC)
      READ (11,800) (XCR(K), K=1, NBC)
      READ (11,800) (YC(K), K=1,NBC)
      READ (11,800) (AREA(K), K=1,NBC)
      READ (11,800) (ARWT(K), K=1, NB1)
      READ (11,800) (SLOPE(K), K=1,NBC)
      DO 100 L=1, NB1
      KS = 1 + (L-1)*NC
      KE = KS + NC1
      READ (11,800) (ALPHAT(K), K=KS, KE)
      ALPHA(KS,1) = ALPHAT(KS)
      ALPHA(KS,2) = -ALPHAT(KS)
      KS1 = KS + 1
      DO 110 K=KS1, KE
      ALPHA(K,1) = SLOPE(K-1) + ALPHAT(K)
      ALPHA(K,2) = SLOPE(K-1) - ALPHAT(K)
  110 CONTINUE
  100 CONTINUE
C
      DO 120 K=1, NBC
      RFAREA = RFAREA + AREA(K)
  120 CONTINUE
  200 CONTINUE
C
      NBC1 = NB1*NC
      DO 350 L=1, 2
      NWG = L
      DO 300 J=1, NB1
      DO 310 I= 1, NC
      IJ = I + (J-1)*NC
      SLOPE(IJ) = -(-1.0)**L*( ALPHA(IJ, L) - ARBTAN*DR)
  310 CONTINUE
  300 CONTINUE
      CALL SECEXP(1)
      DO 320 K=1, NBC1
      CPW(K,L) = CPW(K,L)/BT2
  320 CONTINUE
  350 CONTINUE
      NWG = NWING
C
      CALL FORCEB(X,Y,ARB,YAW,BROLL, BBCL,BBCD, BBCB, BBCM, BBCRM,
     1                BBCYM)
C
      BBCD = BBCD/RFAREA
      BBCL = BBCL/RFAREA
      BBCB = BBCB/RFAREA
      BBCM = BBCM/(RFAREA*CBAR)
      BBCRM = BBCRM/(RFAREA*CBAR)
      BBCYM = BBCYM/(RFAREA*CBAR)
C
      WRITE (6,801)
      WRITE (6,802) BBCD, BBCL, BBCB, BBCM, BBCRM, BBCYM
      WRITE (6,803)
      WRITE (6,804) (THETB(I), I=1,NTHETB)
      WRITE (6,805)
      NBD1 = NBODYS - 1
      DO 360 I=1,NBD1
      WRITE (6,806)  XMIDB(I), ( CPBB(I, J), J=1, NTHETB)
  360 CONTINUE
C
      DO 400 L=1,2
      DO 400 NS=1, NC
  400 CONTINUE
      L = 1
C
      CALL FORCEW(L,NB1,NC1,XC,XCR,YC,AREA,ARB,ALPHA,WCL,WCD,WCB,WCPMU,
     1            WCRMU,WCYMU)
C
      WCLT = WCL
      WCDT = WCD
      WCBT = WCB
C
      L = 2
      CALL FORCEW(L,NB1,NC1,XC,XCR,YC,AREA,ARB,ALPHA,WCL,WCD,WCB,WCPML,
     1            WCRML,WCYML)
C
      WCLT = (-WCLT + WCL)/RFAREA
      WCDT = (-WCDT + WCD)/RFAREA
      WCBT = (-WCBT + WCB)/RFAREA
      WCPM = (-WCPMU + WCPML)/(RFAREA*CBAR)
      WCRM = (-WCRMU + WCRML)/(RFAREA*CWAR)
      WCYM = (-WCYMU + WCYML)/(RFAREA*CBAR)
C
      WRITE (6,807)
      WRITE (6,802) WCDT, WCLT, WCBT, WCPM, WCRM, WCYM
      WRITE (6,808)
      CALL WRITEW( NC1, NB1, CPW(1, 1) )
      WRITE (6,809)
      CALL WRITEW( NC1, NB1, CPW(1, 2) )
C
  800 FORMAT (7F10.4)
  801 FORMAT (1H1,*  PRESSURES, FORCES AND MOMENTS ON ISOLATED BODY*)
  8C2 FORMAT (1H0,*  CD =*,F9.4,5X,*CL =*,F9.4,5X,*CB =*,F9.4,5X,
     1        *CPM =*,F9.4,5X,*CRM =*,F9.4,5X,*CYM =*,F9.4)
  803 FORMAT (1HG,*   BODY PRESSURE COEFFICIENTS (CP) *)
  804 FORMAT (1H0,* THETA(DEG)*,12F10.4/(13X,12F10.4) )
  805 FURMAT (1H0,9X,*X*)
  805 FORMAT (1H ,F12.4,12F10.4/(13X,12F10.4) )
  807 FORMAT (1H1,*  FORCES AND MOMENTS ON WING*)
  8C8 FORMAT (1H0,*   UPPER SURFACE WING PANEL PRESSURE COEFF. (CP)*)
  809 FORMAT (1H0,*   LOWER SURFACE WING PANEL PRESSURE COEFF. (CP)*)
  810 FORMAT (1H0,*   WING PANEL PRESSURE DIFFERENCE (CL) *)
  311 FORMAT (1H0,*   UPPER SURFACE WING PANEL SLOPE*)
  812 FORMAT (1H0,*   LOWER SURFACE WING PANEL SLOPE*)
C
C
      END
      SUBROUTINE CONEAT( XMI, BETA,ALPHA,XMS, PCONE, XMIP, ALFP, THETA)
C
C
C        CALCULATES THE PRESSURE COEFFICIENTS CPT AND PRESSURE AT
C        ANGLE OF ATTACK.
C
      COMMON/CONST /DR, RD, G, GP1, GM1, GR, AV1, AV2, AV3
      COMMON/FLOW  /FLOWFD(10, 2), X(51), Y(51)
      COMMON/PBDATA/NBODYS, XMIDB(51), NTHETB, THETB(15), CPBB(51,15)
      DIMENSION AA(6), A(18), CPX(20), PCONE(1), XMIP(1), ALFP(1),
     1          THETA(1), XMS(1), CPT(20)
C
      DATA  A/ -0.07657, 1.4775, 0.064669, 0.42339,  0.13241, 0.035871,
     1         -0.002085, -0.075797, -0.01923, 0.29898, - 0.10011,
     2         0.29589, -0.99727, -0.41751, 0.048791, -0.039442,
      3          0.10422, 0.063801/
C
      BETAD = BETA * 2.9
      BETAR = BETA * DR
      ALPHAR = ALPHA * DR
      COSB = COS( BETAR )
      SINA = SIN( ALPHAR )
      COSA = COS( ALPHAR )
      SINB = SIN( BETAR )
      CACB = COSA*COSB
      CACB2 = CACB**2
      SASB = SINA*SINB
      PMIN = 0.3
      BBETAR = 2.9 * BETAR
      ALPBET = ABS( ALPHA/BETA )
      T = SIN(BBETAR ) * TAN( BETAR )
      XMI2 = XMI**2
      CALL NCONE( XMI, BETA, CP, DSC)
C
C         DSCD  SHOCK ANGLE FOR ZERO ANGLE OF ATTACK CASE
C
      DSCD = DSC*RD
      SINDS = SIN(DSC)
      CPCONE = ( CP - 1.0 ) / ( 0.5*G*XMI2)
C
C        CALCULATES THE PRESSURE ON CONE (PCONE), FLOW MACH NO.
C       (XMIP), ANGLE OF ATTACK (ALFP) IN MERIDIAN PLANE
C
      DO 200 J=1, NTHETB
      THER = ( 180.0 - THETB(J) )*DR
      COSP = COS(THER)
      COS2P = COS( 2.0*THER)
      SINP = SIN(THER)
      DO 100 I = 1, 6
      K = 3*(I-1) + 1
      AA(I) = A(K) + A(K+1)*COSP +A(K+2)*COS2P
  100 CONTINUE
C
      CPX(J) = ( AA(1)*T + AA(2)*T/XMI2 + AA(3)/XMI2 )*ALPBET +
     1        ( AA(4)*T + AA(5)*T/XMI2 + AA(6)/XMI2 )*ALPBET**2
      CPT(J) = CPX(J) + CPCONE + CPSTAG
      PCONE(J) = 0.5*G*XMI2*CPT(J) + 1.0
      IF ( PCONE(J) .LT. PMIN ) PCONE(J) = PMIN
      XMIP(J) = XMI*SQRT( COSA**2 + (SINA*COSP)**2 )
      ALFP(J) = ATAN( SINA*COSP/COSA ) * RD
      IF ( ABS(ALPHA) .LT. 1.0E-2 ) GO TO 201
  200 CONTINUE
  201 CONTINUE
C
C         CALCULATES THE SURFACE MACH NO. (XMS), ( SHOCK ANGLE -
C        DEFLECTION ANGLE )  (THETA)
C
      DO 300 J=1, NTHETB
      DELI = BETAR
      IS = 1
      FDP = 1.0
      XMC = XMIP(J)
      GO TO 400
  450 CONTINUE
      XMNS = (XMI*SINDS)**2
      PS = (2.0*G*XMNS - GM1)/GP1
      FDP = PS/P2
      XMC = XMIP(J)
      DELI = ALFP(J)*DR + BETAR
      DPSK = 0.0
      IF ( DELI .LE. 0.0 ) GO TO 500
      IS = 2
  460 CONTINUE
      DELID = DELI*RD
      XMD = XMC*SIN(DELI)
      CK = 2.0*GP1/(G +3.0)*XMD
      XMNS = ( CK + EXP(-CK))**2
      P2 = ( 2.0*G*XMNS - GM1 )/ GP1
      GO TO ( 450, 470), IS
  470 CONTINUE
      DPSK = PCONE(J) - P2*FDP
  500 IF ( DPSK .LT. 0.0 ) DPSK = 0.0
      PSHK = PCONE(J) - DPSK
      IF ( PSHK .LT. 1.0 ) PSHK = 1.0
      XMI2 = XMIP(J)**2
      XMS(J) =SQRT((XMI2*(GP1*PSHK + GM1 ) - 2.0*GP1*GM1 )/(PSHK*(
     1          GM1*PSHK + GP1) ) )
      EM2 = XMIP(J)**2
      SINBT2 = (GP1*PSHK + GM1)/(2.0*G*EM2)
      IF ( SINBT2 .GT. 1.0 ) SINBT2 = 1.0
      IF ( SINBT2 .LT. 0.0 ) SINBT2 = 1.0/EM2
      SHOCKA = ASIN( SQRT(SINBT2) )
      THETA(J) = SHOCKA*RD - ALFP(J)
      IF ( ABS(ALPHA) .LT. 1.0E-2 ) GO TO 301
  300 CONTINUE
  301 CONTINUE
C     WRITE(6,801)(THETB(I),CPT(I),PCONE(I),XMIP(I),XMS(I),THETA(I),
C    1         ALFP(I),I=1,NTHETB)
C 801 FORMAT (1H0,20X,*FLOW FIELD IN THE MERIDIANAL PLANE*///
C    1         5X,*MERIDIAN                 CP              P/PINF          *,
C    2         3X,*M1                 M2          SHOCK ANGLE    ANGLE OF*/
C    3         5X,*  ANGLE*,86X,*ATTACK*//(3X,F9.2,3X,6F15.4))
C
C
      RETURN
      END
      SUBROUTINE NCONE( EM, DCD, PCP1, DSR)
C
C
C        CALCULATES THE CONE PRESSURE AT ZERO ANGLE OF ATTACK
C        SECOND ORDER THEORY USED FOR SMALL SIMILSRITY PARAMETER
C        APPROXIMATE SOLUTION OF HAMMITT AND MURTHY FOR LARGE VALUES
C        MIDRANGE GIVEN BY SUITABLE TRANSITION FUNCTION
C
      COMMON/CONST /DR, RD, G, GP1, GM1, GR, AV1, AV2, AV3
      ISDET = 0
      RC = 0.017453292
      DCR = DCD * DR
      SINDC = SIN(DCR)
      GM12 = 0.5*GM1
      EMSIN = EM * SINDC
      EMSQ = EM**2
      KSOL = 2
C
C
C     CALCULATE UPPER TRANSITION POINT, EMSINF
C
      EMSINF = 1.4
      IF ( EM  .GE. 10.0 ) GO TO 8
      EMSINF = 0.325
      IF ( EM  .GT. 1.5 ) EMSINF = 1.4 - 1.075*EXP((EM - 1.5)*(-0.8) )
    8 IF ( EMSIN  .LT. EMSINF ) GO TO 20
      KSOL = 1
C
C        CALCULATE CONE SURFACE PROPERTIES USING METHOD OF HAMMITT AND
```

```
C        MURTHY
C
  9   CONTINUE
      SINSQ = SINDC**2
      SIN2 = SIN( 2.0*DCR )
      N1 = 2.0 - (G+5.0) * SINSQ
      N2 = GM1 *SINSQ + 2.0/EMSQ
      NX = N2 * N1 / SIN2**2
      IF ( ABS( N1 )  .LE. 1.E-3 ) GO TO 10
      RAD = SQRT( 1.0 + NX )
      DSR = DCR - SIN2*( 1.0 - RAD ) / N1
      GO TO 11
 10   CONTINUE
      DSR = DCR + 0.5*N2/SIN2*(1.0 - 0.25*NX*(1.0-0.5*NX) )
 11   CONTINUE
      EMNSSQ = EMSQ * SIN(DSR)**2
      NS2 = ( DSR - DCR ) **2
      EMC = SQRT( ( EMSQ - EMNSSQ )*(1.0 + 2.0*NS2)/(1.0 + GM12*
     1        (EMNSSQ*(1.0 + 2.0*NS2) -2.0*NS2*EMSQ) ) )
      ELIM = GM1/GP1
      PSP1 = 2.0 * G / GP1* EMNSSQ - ELIM
      T1TS = ( PSP1 + ELIM) / ( 1.0 + ELIM*PSP1) / PSP1
      TCT1 = ( 1.0 + GM12 *EMSQ)/( 1.0 + GM12*EMC**2 )
      PCPS = ( TCT1*T1TS)**( G / GM1 )
      PCP1 = PCPS * PSP1
      CP = 2.0*( PCP1 - 1.0 ) / ( G*EMSQ)
      GO TO (40,31), KSOL
C
C        CALCULATE LOWER TRANSITION POINT, EMSINO
C
  20  CONTINUE
      EMSINO = 0.2
      IF ( EM  .GT. 3.0 ) EMSINO = 0.3
      IF ( EMSIN  .GT. EMSINO ) GO TO 30
      KSOL = 1
C
C        CALCULATE CONE SURFACE PROPERTIES USING SECOND ORDER THEORY
C
 21   CONTINUE
      BETA = SQRT( EMSQ - 1.0 )
      TANDC = TAN( DCR )
      TANSQ = TANDC**2
      BETA2 = 5.0*EMSQ - 1.0
      AM = 3.25*EMSQ + 0.5 + GP1*( EMSQ/BETA)**2
      PHI = 0.69314718 - ALOG(BETA*TANDC)
      CP = TANSQ*(2.0*PHI-1.0+TANSQ*(3.0*(BETA*PHI)**2-BETA2*PHI+AM) )
      IF ( KSOL  .EQ. 2 ) GO TO 22
      DSR =   ASIN( 1.0 / EM )
      PCP1 = G * EMSQ * CP * 0.5 + 1.0
      TCT1 = PCP1 **(GM1 / G )
      EMC = SQRT( 2.0 / GM1*( (1.0+GM12*EMSQ)/TCT1 -1.0 ) )
      GO TO 40
 22   CONTINUE
      DCP2 = TANDC*( 4.0*(PHI - 1.0) + TANSQ*(6.0*PHI*(2.0*PHI-1.0)*
     1          BETA**2-BETA2*(4.0*PHI -1.0) + 6.0*AM))/(EM*COS(DCR)**3)
      GO TO 32
C
C        CALCULATE CONE SURFACE PROPERTIES USING TRANSITION SOLUTION
C
 30   CONTINUE
      DXF = EMSINF- EMSINO
      DX = EMSIN - EMSINO
      SINDC = EMSINF / EM
      DCR =  ASIN( SINDC )
      GO TO 9
```

```
   31 CONTINUE
      CPHM = CP
      DCR =  ASIN( EMSINO/ EM )
      GO TO 21
   32 CONTINUE
      A = ( CPHM - CP - DCP2*DXF ) / DXF**2
      CP = CP + DX * ( DCP2 + A*DX )
      IF ( ISDET  .EQ. 2 ) RETURN
      PCP1 = 0.5* G * EMSQ * CP + 1.0
      EMNSF = EM * SIN(DSR)
      EMNS = ( EMNSF - 1.0) * ( EMSIN / EMSINF)**2 + 1.0
      EMNSSQ = EMNS **2
      EMSSQ = (GP1*EM*EMNS)**2 - 4.0*(EMNSSQ-1.0)*(G *EMNSSQ + 1.0)
      DSR =  ASIN( EMNS / EM )
      EMSSQ = EMSSQ /((2.0*G*EMNSSQ -GM1)*(GM1*EMNSSQ + 2.0))
      PSPC = (2.0*G*EMNSSQ - GM1)/GP1/ PCP1
      EMC = SQRT( ( (1.0+GM12*EMSSQ)*PSPC**(GM1/G) -1.0)/GM12)
C
      TCT1 = ( 1.0+GM12*EMSQ) /( 1.0+GM12*EMC**2)
   40 CONTINUE
C
      RETURN
      END
      SUBROUTINE COMP( DANG, CP, SKANG)
C
C      COMPUTES THE CONDITION BEHIND THE SHOCK
C
      COMMON/CONST /DR, RD, G, GP1, GM1, GR, AV1, AV2, AV3
      COMMON/FLOW  /FLOWFD(10, 2), X(51), YY(51)
C
C
C          CHECK THE DEFLECTION ANGLE DANG
C
      XMI = FLOWFD(1,1)
      XMI2 = XMI**2
      IF ( ABS(DANG) .LT. 1.0E-5 ) GO TO 240
      IF ( DANG .GT. 55.0 ) GO TO 10
      IF ( ABS(DANG) .LE. 2.0 ) GO TO 260
C
C         SET UP CUBIC AND SOLVE FOR SIN(THETA)**2 OR SHOCK ANGLE
C
      SIND = SIN(DANG*DR)
      COSD = COS(DANG*DR)
      B = -(XMI2 + 2.0)/XMI2 - G*SIND**2
      C = (2.0*XMI2 + 1.0)/XMI2**2+(GP1**2/4.0 +GM1/XMI2 )*SIND**2
      D = -( COSD / XMI2 )**2
C
C         CHECK FOR SHOCK DETACHMENT
C
      CHECK = (-B**2/9.0 + C/3.0 )**3 + ( (B/3.0)**3 - (B*C -3.0*D)
     1      /6.0)**2
      IF ( CHECK .GE. 0.0 ) GO TO 10
C
C         ATTACHED SHOCK
C
      Y = B**2 - 3.0*C
      Z = (9.0*B*C - 2.0*B**3 - 27.0*D)/(2.0*Y**1.5)
      W = ACOS(Z)
      Z = W/3.0
      Y = 2.0*SQRT(Y)
      R1 = (Y*COS(Z) - B)/3.0
      R2 = -(Y*COS(Z + 60.0*DR ) + B)/3.0
      R3 = -(Y*COS(Z - 60.0*DR ) + B)/3.0
      GO TO 20
C
```

```
C         NO CUBIC ROOT WAS FOUND, SHOCK DETACHED
C
   10 CONTINUE
      WRITE (6, 800)
C
C         CUBIC SOLUTION FOUND, ONLY THE MIDDLE ROOT IS REALISTIC
C         AND WILL BE USED
C
   20 CONTINUE
      IF ( R1 - R2 ) 30, 110, 40
   30 K = 1
      GO TO 50
   40 K = 2
   50 IF ( R2 - R3 ) 60, 120, 70
   60 M = 1
      GO TO 80
   70 M = 2
   80 IF ( K .EQ. M ) GO TO 120
      IF ( R1 - R3 ) 90, 130, 100
   90 GO TO ( 130, 110), K
  100 GO TO ( 110, 130), K
  110 SKANG2 = R1
      GO TO 140
  120 SKANG2 = R2
      GO TO 140
  130 SKANG2 = R3
C
C         CHECK IF SHOCK ANGLE IS NEGATIVE AND PRINT
C
  140 CONTINUE
      IF ( SKANG2 .GT. 0.0 ) GO TO 170
      WRITE (6,805) SKANG2
      SKANG2 = 0.0
  170 IF ( SKANG2 .LE. 1.0 ) GO TO 200
      WRITE (6,810) SKANG2
      SKANG2 = 1.0
C
  200 CONTINUE
      XMSIN2 = XMI2 * SKANG2
  210 CONTINUE
      IF ( XMSIN2 .LT. 1.01 ) XMSIN2 = 1.01
      FLOWFD(4,2) = FLOWFD(4,1)*GP1*XMSIN2/(GM1*XMSIN2 + 2.0)
      FLOWFD(3,2) = FLOWFD(3,1)*(2.0*G*XMSIN2 - GM1)/GP1
      FLOWFD(5,2) = FLOWFD(5,1)*(2.0*G*XMSIN2 - GM1)*(2.0*GM1*XMSIN2)
     1        /( GP1**2*XMSIN2)
      FLOWFD(6,2) = FLOWFD(6,1) * SQRT( FLOWFD(5,2)/FLOWFD(5,1) )
      IF ( FLOWFD(5,2) .GE. AV1 ) FLOWFD(7,2)= 2.27E-8 * FLOWFD(5,2)**1.5
     1        /( FLOWFD(5,2) + 198.6 )
      IF ( FLOWFD(5,2) .LT. AV1 ) FLOWFD(7,2) = AV2*FLOWFD(5,2)**AV3
      FLOWFD(1,2) = ( GP1**2*XMI2*XMSIN2 - 4.0*(XMSIN2 - 1.0)*(G*XMSIN2+
     1          1.0) )/( (2.0*G*XMSIN2 - GM1)*( 2.0 + GM1*XMSIN2 ) )
      FLOWFD(1,2) = SQRT( FLOWFD(1,2) )
      IF ( FLOWFD(1,2) .LT. 1.0 ) FLOWFD(1,2) = 1.01
      FLOWFD(2,2)  = FLOWFD(1,2) * FLOWFD(6,2)
      FLOWFD(8,2) = FLOWFD(4,2) * FLOWFD(2,2) / FLOWFD(7,2)
      SKANG = SQRT(SKANG2 )
      IF ( ABS( SKANG ) .GT. 1.0 ) SKANG = 1.0
      SKANG = ASIN( SKANG ) * RD
C
      CP = 4.0*(XMSIN2 - 1.0)/( GP1*XMI2 )
C
  230 RETURN
C
C         ANGLE OF DEFLECTION EQUALS TO ZERO, CONTINUE THE FLOW FIELD
C
```

```
  240 CONTINUE
      DO 250 I=1, 8
      FLOWFD(I,2) = FLOWFD(I,1)
  250 CONTINUE
      CP = 0.0
      RETURN
C
C        WEAK OBLIQUE SHOCK RELATIONSHIP
C
  260 CONTINUE
      SKANG2 = (1.0 + 0.5*GP1/SQRT(XMI2 - 1.0)*DANG*DR ) / XMI2
      GO TO 200
C
  800 FORMAT (1H0,10X,*NO CUBIC ROOT, DETACHED SHOCK*)
  805 FORMAT (1H0,10X,*NEGATIVE SHOCK ANGLE, SINE OF SHOCK ANGLE =*,
     1       F12.3)
  810 FORMAT (1H0,10X,*SINE OF SHOCK ANGLE .GT. 1.0*,F12.3)
C
C
      END
      SUBROUTINE EXPAND( DANG, CP)
C
C
C       PRANDTL - MEYER EXPANSION ( DANG .GT. 0 ) OR COMPRESSION
C      ( DANG .LT. 0 ) FOR A GIVEN FREE STREAM CONDITIONS
C
      COMMON/CONST /DR, RD, G, GP1, GM1, GR, AV1, AV2, AV3
      COMMON/FLOW  /FLOWFD(10, 2), X(51), Y(51)
C
      DIMENSION A(2), C(2)
C
C
      XMI = FLOWFD(1,1)
C
      IF ( FLOWFD(1,1) .GE. 1.0 ) GO TO 10
C
C         FLOW SUBSONIC, SET MACH NO AT UNITY FOR CONTINUATION
C
      XMI2 = 1.0
      GO TO 20
C
   10 CONTINUE
      XMI2 = XMI**2
C
   20 CONTINUE
C
C
C         CALCULATE THE PRANDTL-MEYER ANGLE NU1 (RADIAN) FOR FREE
C        STREAM
C
      XNU1 = GR*ATAN( SQRT(XMI2 - 1.0)/GR ) - ATAN( SQRT(XMI2 - 1.0))
C
C         CALCULATE THE PRANDTL-MEYER ANGLE NU2 AFTER TURNING
C
      XNU1D = XNU1*RD
      XNU2D = XNU1D + DANG
      XNU2 = XNU2D*DR
C
      IF ( XNU2D .GT. 0.0 ) GO TO 30
C
C         FLOW AFTER TURNING IS SUBSONIC
C
      XM2A = 1.0
      GO TO 190
C
```

```
   30 CONTINUE
      CHECKM = 2.2768532
      IF ( XNU2  .LT. CHECKM ) GO TO 40
C
C          FLOW EXPANDS TO INFINITE MACH NUMBER, SET TO 100.0
C
      XM2A = 100.0
      GO TO 190
C
C          START OF ITERATION FOR THE MACH NUMBER AFTER TURNING
C
   40 CONTINUE
C
      I = 1
      EPS = 1.0E-6
C
C          BEGIN WITH THE APPROXIMATE MACH NUMBER XM2A
C
      XM2A = XMI*(1.0 + (XNU2-XNU1)*(1.0 + 0.5*GM1*XMI2)/
     1       SQRT(XMI2 - 1.0) )
      IF ( ABS( DANG ) .LT. 1.0E-1 ) GO TO 190
C
   50 CONTINUE
C
      TM1 = ( XNU2 + ATAN( SQRT( XM2A**2 - 1.0 ) ) )/ GR
      TM2 = TAN ( TM1 )
      XM2 = SQRT( 1.0 + ( TM2*GR)**2 )
      IF ( I .GT. 1 ) GO TO 60
      I = I + 1
      DM1 = XM2 - XM2A
      XM2B = XM2A
      XM2A = 1.1*XM2A
      GO TO 50
   60 CONTINUE
      I = I + 1
      DM2 = XM2 - XM2A
      IF ( ABS( DM2/XM2A ) .LT. EPS ) GO TO 190
      IF ( I .GT. 20 ) GO TO 200
      XM2C = XM2A
      XM2A = XM2A - DM2*(XM2A-XM2B)/(DM2-DM1)
      XM2B = XM2C
      DM1 = DM2
      GO TO 50
C
C
C          ITERATION COMPLETED, CALCULATE FLOW FIELD AFTER TURNING
C
  190 CONTINUE
      FLOWFD(1,2) = XM2A
      ONEOM = 1.0/XM2A
      SKANG = ASIN(ONEOM)*RD
      Z = (2.0 + GM1*XMI2)/(2.0 + GM1*XM2A**2 )
      FLOWFD(3,2) = FLOWFD(3,1) * Z**(G/GM1)
      FLOWFD(4,2) = FLOWFD(4,1) * Z**(1.0/GM1)
      FLOWFD(5,2) = FLOWFD(5,1) * Z
      FLOWFD(6,2) = FLOWFD(6,1) * SQRT( Z )
      IF ( FLOWFD(5,2) .GE. AV1 ) FLOWFD(7,2)=2.27E-8*FLOWFD(5,2)**1.5
     1       /( FLOWFD(5,2) + 198.6)
      IF ( FLOWFD(5,2) .LT. AV1 ) FLOWFD(7,2) = AV2*FLOWFD(5,2)**AV3
      FLOWFD(2,2) = XM2A*FLOWFD(6,2)
      FLOWFD(8,2) = FLOWFD(4,2)*FLOWFD(2,2)/FLOWFD(7,2)
      CP = (Z**(G/GM1) - 1.0)/(0.5*G*XMI2)
C
      GO TO 300
  200 CONTINUE
```

```
      WRITE (6,800) I, XM2A, XM2
C
  300 CONTINUE
C
  800 FORMAT (1H0,*ITERATION FAIL TO CONVERGE AFTER *,I5,2X,*TIMES*,
     1          * CALCULATION CONTINUES*,2F15.3)
C
      RETURN
      END
      SUBROUTINE SECEXP(IBW)
C
C
C         SECOND ORDER SHOCK EXPANSION
C           IBW = 1, 2-D FLOW
C                 2, CONICAL FLOW
C
      COMMON/CONST /DR, RD, G, GP1, GM1, GR, AV1, AV2, AV3
      COMMON/FREES /XMACH, PF, ALPHA, BETA, NEND
      COMMON/FLOW  /FLOWFD(10, 2), XS(51), Y(51)
      COMMON /NVAR/  NDUM(14), NROWB, NROWW, NCOLW
      COMMON/PBDATA/NBODYS, XNIDB(51), NTHETB, THETB(15), CPBB(51,15)
      COMMON/XVAR  /DUMMY(426), XB(51)
      COMMON/WINGOT/ CPW,NWG,ISOLID,IFORM(10),YAW,BROLL,SLOPE(51)
C
      DIMENSION CPW(110, 2)
C
C
C
C         MACH NUMBER - FLOWFD(1,1)
C         VELOCITY - FLOWFD(2,1)
C         PRESSURE - FLOWFD(3,1)
C         DENSITY - FLOWFD(4,1)
C         TEMPERATURE - FLOWFD(5,1)
C         SPEED OF SOUND - FLOWFD(6,1)
C         VISCOSITY - FLOWFD(7,1)
C         REYNOLDS NUMBER - FLOWFD(8,1)
C
      FLOWFD(1,1) = XMACH
      FLOWFD(3,1) = 1.0
      FLOWFD(4,1) = 1.0
      FLOWFD(5,1) = 520.0
      FLOWFD(6,1) = 49.0*SQRT(FLOWFD(5,1))
      FLOWFD(2,1) = FLOWFD(1,1) * FLOWFD(6,1)
      IF ( FLOWFD(5,1) .GE. AV1 ) FLOWFD(7,1)=2.27E-8*FLOWFD(5,1)**1.5
     1       /( FLOWFD(5,1) + 198.6)
      IF ( FLOWFD(5,1) .LT. AV1 ) FLOWFD(7,1) = AV2*FLOWFD(5,1)**AV3
      FLOWFD(8,1) = FLOWFD(4,1)*FLOWFD(2,1)/FLOWFD(7,1)
C
C         CALCULATE PRESSURE ON THE BODY
C
      GO TO ( 200, 100), IBW
C
  100 CONTINUE
      CALL BODYP
      GO TO 300
C
C         CALCULATE PRESSURE ON THE WING
  200 CONTINUE
C
      NB1 = NCOLW - 1
      DO 250 I=1, NB1
      FLOWFD(1, 1) = XMACH
      FLOWFD(3, 1) = 1.0
      CALL WINGP(I)
  250 CONTINUE
  300 CONTINUE
C
C
C
      RETURN
      END
      SUBROUTINE WINGP(IB)
C
C
C        CALCULATES PRESSURE ON THE WING ( CPW ) FOR GIVEN SURFACE
C       SLOPE (SLOPE) AND ANGLE OF ATTACK
C
      COMMON/CONST /DR, RD, G, GP1, GM1, GR, AV1, AV2, AV3
      COMMON/FREES /XMACH, PF, ALPHA, BETA, NEND
      COMMON/FLOW  /FLOWFD(10, 2), XS(51), Y(51)
      COMMON /NVAR/  MDUM(14), NROWB, NC, NB
      COMMON/WINGOT/ CPW,NWG,ISOLID,IFORM(10),TAW,BROLL,SLOPE(51)
C
      DIMENSION CPW(110,2), XMID(51), CPMID(51), THETA(51)
C
      NB1 = NB - 1
      NCS = (IB -1)*NC
      IS1 = NCS + 1
      PMIN = 0.3
      GR1 = GM1 / G
C
C       TWO DIM. STARTING FLOW
C
      ISE = NC + NCS
C
      DANG =  SLOPE(IS1)*RD
      IF ( DANG .LT. 0.0 ) GO TO 200
C
      CALL COMP( DANG, CP, SKANG)
      THETA(1) = SKANG
      GO TO 210
C
  200 CONTINUE
      DANG = -DANG
      CALL EXPAND( DANG, CP)
C
      CPMID(1) = 0.
  210 CONTINUE
      CPW(IS1, NWG)   FLOWFD(3,2) - 1.0
      XMID(IS1) = FLOWFD(1,2)
      CPMID(IS1+1) = CP
      THET  = SKANG - ALPHAD
      THETAR = THET *DR
      TANT = TAN(THETAR)
C
      DO 590 I=2,NC
      IS = I + NCS
      DANG = RD*( SLOPE(IS) - SLOPE(IS-1) )
      FLOWFD(1,1) = XMID(IS-1)
      FLOWFD(3,1) = CPW(IS-1, NWG) + 1.0
      IF ( DANG .LT. 0.0 ) GO TO 525
      CALL COMP( DANG, CP, SKANG)
      GO TO 570
C
  525 CONTINUE
      DANG = - DANG
      CALL EXPAND( DANG, CP)
  570 CONTINUE
      CPW(IS, NWG) = FLOWFD(3,2) - 1.0
      XMID(IS) = FLOWFD(1,2)
      CPMID(IS+1) = ( CP + 1.0 )*( CPMID(IS) + 1.0 ) - 1.0
  590 CONTINUE
  600 CONTINUE
C
      RETURN
      END
      SUBROUTINE BODYP
C
C
C         CALCULATE THE PRESSURE ON THE BODY ( CPBB ) AT XMIDB AND
C      THETB
C
      COMMON / BODYSP/ DRDX(51), DZDX(51), DRDXC(51),XWROOT,XWTIP
      COMMON/CONST /DR, RD, G, GP1, GM1, GR, AV1, AV2, AV3
      COMMON/FREES /XMACH, PF, ALPHA, BETA, NEND
      COMMON/FLOW  /FLOWFD(10, 2), XS(51), Y(51)
      COMMON/PEDATA/NBODYS, XMIDB(51), NTHETB, THETB(15), CPBB(51,15)
      DIMENSION PCONE(15, 2), XMIP(15), ALFP(15), THETA(15),
     1          XMS(15), S(75), PS(15), RS(51), DPDS(15),XMID(2, 15)
C
C
      PMIN = 0.3
      IC = 1
      ICHECK = 0
      XMI = XMACH
      BT2 = 0.5*G*XMACH**2
      DPDS3 = 0.0
      XS(1) = 0.0
      RS(1) = 0.0
      GR1 = GM1 / G
C
C         CONICAL STARTING FLOW
C
      BETA = ATAN( DRDX(1) )*RD
      CALL CONEAT( XMI, BETA,ALPHA,XMS(1),PCONE(1,1),XMIP,ALFP,
     1           THETA)
      CONANGP = BETA
      RS(2) = RS(1) + (XS(2)-XS(1))*TAN(THETA(4)*DR)
      DO 275 J=1,NTHETB
      PS(J) = PCONE(J, 1)
      CPBB(1, J) = PCONE(J, 1)
      DPDS(J) = DPDS3
  275 CONTINUE
C
      PSHK = (2.0*G*SINM1 - GM1)/GP1
      DPSH= PC - PSHK
      IC = 2
C
C        CONICAL FLOW
C
      DO 450 I=3,NBODYS
      I2 = 2*(I-2) - 1
      I3 = I2 + 1
      I4 = I3 + 1
      RADI = Y(I-1)
      CONANG = RD*ATAN(DRDX(I-1) )
      DANG = CONANG - CONANGP
      DS = SQRT( (XMIDB(I-1)-XMIDB(I-2))**2 + (Y(I)-Y(I-1))**2 )
C
      IF ( ABS(DANG) .LT. 0.1 .OR. CONANG .LT. 0.1 ) GO TO 285
C
      CALL CONEAT(XMI,CONANG,ALPHA,XMS(1),PCONE(1, 2),XMIP,ALFP,
     1           THETA)
C
C         CALCULATE PRESSURE ON CONE (PA) IN THE MERIDIANAL PLANE
```

```
  285 CONTINUE
C
      DO 310 J=1, NTHETB
      IF ( I .GT. 3 ) GO TO 405
      XM1 = XMS(J)
      P1 = CPBB(1, J)
      PC = PCONE(J, 2)
      PS(J) = PCONE(J, 1)
      GO TO 407
  405 CONTINUE
      XM1 = XMS(J)
      XMID(1, J) = XMID(2, J)
      P1 = PS(J)
      PC = PCONE(J, 2)
      DPDS1 = DPDS(J)
  407 CONTINUE
      XNU1 = GR*ATAN( SQRT(XM1**2 - 1.0)/GR ) - ATAN(SQRT(XM1**2 -
     1      1.0) )
      XNU2 = XNU1 - DANG*DR
      IF ( XNU2 .GT. XNU1 ) GO TO 410
C
      IF ( DANG .LT. 1.0 ) GO TO 410
C
C          SHOCK CALCULATION
C
      FLOWFD(1, 1) = XM1
      FLOWFD(3, 1) = P1
      CALL COMP( DANG, CP, SKANG)
C
      SKANGR = SKANG*DR
      XM2 = FLOWFD(1,2)
      P2 = FLOWFD(3,2)
      TAND = TAN( DANG*DR )
      COSS = COS( SKANGR )
      SINS = SIN( SKANGR )
      SINSA = SIN( (SKANG - DANG )*DR )
      SINM = SIN( 1.0/XM2 )
      XMU2 = ASIN( 1.0/ XM2 )
      TANM = TAN(XMU2)
      TM1 = (GP1*TAND*COSS - SINS)*(XM1*SINS)**2 + SINS
      TM2 = 1.0 + (1.0 -2.0*SINS**2 + 2.0*TAND*SINS*COSS)*(XM1*SINS)
     1       **2
      TM3 = (1.0 + 0.5*GM1*XM1**2)*SINS*TM1/(GP1*TM2/4.0)
      B1 = G*P1*XM1**2/(2.0*(XM1**2 - 1.0))
      B2 = G*P2*XM2**2/(2.0*(XM2**2 - 1.0))
      TM4 = 2.0*B2/RADI*(SINS*SIN(CONANGP*DR)/SINSA - SIN(CONANG*DR))
     1       + DPDS1*(B2*SINS/(B1*SINSA) + (P2/P1-TM3)*COSS*TANM/SINSA)
      DPDS2= TM4 / ( 1.0 + TANM/TAN( (SKANG - DANG)*DR) )
      GO TO 420
C
  410 CONTINUE
C
C          PRANDTL-MEYER EXPANSION
C
      DANGI = -DANG
      FLOWFD(1,1) = XM1
      FLOWFD(3,1) = P1
      CALL EXPAND( DANGI, CP )
      XM2 = FLOWFD(1,2)
      P2 = P1*( 0.5*CP*G*XM1**2 + 1.0)
      IF ( P2 .LT. PMIN ) P2 = PMIN
      IF ( ABS(DANG) .LT. 0.1 .AND. ICHECK .EQ. 0 ) GO TO 430
      OMEG12 = XM2/XM1*((1.0 + 0.5*GM1*XM1**2)/(1.0 + 0.5*GM1*XM2**2))
     1          **(0.5*GR**2)
      B1 = 0.5*G*P1*XM1**2/(XM1**2 - 1.0)
      B2 = 0.5*G*P2*XM2**2/(XM2**2 - 1.0)
      DPDS2= B2/Y(I-1)*(OMEG12*SIN(CONANGP*DR) - SIN(CONANG*DR) ) +
     1        B2/B1*OMEG12*DPDS1
C
  420 CONTINUE
      IF ( PC .EQ. P2 ) GO TO 430
      ETA = DPDS2* DS/(PC - P2)
      IF ( ETA .GT. 180.0 ) ETA = 180.0
      IF ( ETA ) 430, 430, 440
C
  430 CONTINUE
      XM3 = XM2
      P3 = P2
      CPBB(I-1,J) = P3
      XMID(2, J) = XM3
      DPDS3 = 0.0
      GO TO 400
C
  440 CONTINUE
      P3 = PC - (PC - P2)*EXP(-ETA)
      XM3 = SQRT(2.0/GM1*((1.0 + 0.5*GM1*XM2**2)*(P2/P3)**GR1 - 1.0))
      ETAMID = 0.5*ETA
      CPBB(I-1,J) = PC - (PC-P2)*EXP(-ETAMID)
      XMID(2, J) = SQRT(2.0/GM1*((1.0 + 0.5*GM1*XM2**2)*(P2
     1             /CPBB(I-1,J))**GR1 - 1.0) )
      DPDS3 = (PC - P3)/(PC - P2)*DPDS2
  400 CONTINUE
C
      XMS(J) = XM3
      PS(J) = P3
      DPDS(J) = DPDS3
C
      IF ( ABS( ALPHA ) .LT. 1.0E-2 ) GO TO 311
  310 CONTINUE
      GO TO 350
  311 CONTINUE
      DO 315 J=2, NTHETB
      CPBB(I-1, J) = CPBB(I-1, 1)
  315 CONTINUE
C
  350 CONTINUE
C
      RS(I) = RS(I-1) + (XS(I)-XS(I-1))*TAN( THETA(4)*DR )
C
  375 CONTINUE
      IC = IC + 1
C
      CONANGP = CONANG
C
      IF ( ABS(DANG) .LT. 0.1 .AND. ICHECK .EQ. 1 ) GO TO 450
      ICHECK = 1
      RAT = ( XWROOT - XS(I-1) )/( XS(I) - XS(I-1) )
      XMROOT = XMID(1,4) + ( XMID(2,4) - XMID(1,4) ) * RAT
      PROOT = CPBB(I-2,4) + ( CPBB(I-1,4) - CPBB(I-2,4) ) * RAT
      RROOT = Y(I-1) + ( Y(I) - Y(I-1) ) * RAT
C     RSHK1 = RS(I-1) + ( RS(I) - RS(I-1) ) * RAT
  450 CONTINUE
C
      DO 500 I=1, NBODYS
      DO 500 J=1, NTHETB
      CPBB(I, J) = (CPBB(I, J) - 1.0)/DT2
  500 CONTINUE
C
C
C
```

```
      RETURN
      END
      SUBROUTINE FORCEB( XB, RB, ARB, YAW, BROLL,BBCL, BBCD, BBCB,
     1              BBCM,BBCRM, BBCYM)
C
C          INTEGRATES PRESSURE (CPBB) ON THE BODY FOR FORCES
C
      COMMON/CONST /DR, RD, G, GP1, GM1, GR, AV1, AV2, AV3
      COMMON / BODYSP/ DRDX(51), DZDX(51), DRDXC(51),XWROOT,XWTIP
      COMMON/PBDATA/NBODYS, XNIDB(51), NTHETB, THETB(15), CPBB(51,15)
      COMMON/INPUT / TX(2,  50)
      COMMON/ POINTP/ TY(4), TYP(4), TYPP(4)
      COMMON/ POINTS/ NPTS, NVAR
      DIMENSION CP(15), CPY(15), CPZ(15), CPS(51), CPYS(51),
     1          CPZS(51), COSB(15), SINB(15), XB(1), RB(1)
C
      PI=3.1415926
      NT = NTHETB
      NT1 = NT + 1
      NT2 = 2*( NT - 1 )
      NT3 = NTHETB
      NT4 = NT2 + 1
      DX = XB(2) - XB(1)
      DTHETB = (THETB(2) - THETB(1))*DR
      TANA = TAN(ARB*DR)
      COSA = COS(ARB*DR)
      SINA = SIN(ARB*DR)
      TANY = TAN(YAW*DR)
      A1 = 1.0/SQRT( 1.0 + TANA**2 + TANY**2 )
      COSBT = A1 * SQRT( 1.0 + TANA**2 )
      SINBT = SQRT( 1.0 - COSBT**2 )
      COSG = COS(BROLL*DR)
      SING = SIN(BROLL*DR)
      NBD1 = NBODYS - 1
      FROLL = 0.0
      DO 60 I=1, NT4
      THETB(I) = (I-1)*DTHETB/DR
      COSB(I) = COS( THETB(I)*DR )
      SINB(I) = SIN( THETB(I)*DR )
   60 CONTINUE
C
C          INCORPORATE EFFECT OF YAW TO ANGLE OF ATTACK
C
      ARAYAW = ACOS( A1 )
      SINAY = SIN(ARAYAW)
      IF ( YAW .LT. 0.01 .AND. BROLL .LT. 0.01) GO TO 190
      NT = NT2
      NT3 = NT + 1
      IF ( ARB .LT. 0.01 ) GO TO 90
      FROLL = ATAN( TANY / TANA ) * RD
      GO TO 150
   90 CONTINUE
      FROLL = 90.0
  150 CONTINUE
C
C          SPLINE FIT CPBB IN THE BODY COORDINATE SYSTEM
C
      DO 160 I=1, NT
      CPS(I) = THETB(I) + FROLL - BROLL
      IF ( CPS(I) .GT. 180.0) CPS(I) = ABS( 360.0 - CPS(I) )
      IF ( CPS(I) .LT. 0.0 ) CPS(I) = ABS( CPS(I) )
  160 CONTINUE
      NVAR = 2
      NPTS = NTHETB
      DO 180 J=1, NBD1
      DO 170 I=1, NTHETB
      TX(1, I) = THETB(I)
      TX(2, I) = CPBB(J, I)
  170 CONTINUE
      CALL SPLFIT
      DO 175 I=1, NT
      TY(1) = CPS(I)
      CALL POINT(1)
      CPBB(J, I) = TY(2)
  175 CONTINUE
      CPBB(J,NT3) = CPBB(J, 1)
  180 CONTINUE
      NTHETB = NT
C
  190 CONTINUE
      DO 200 J=1, NBD1
      J1 = J + 1
      RM = RB(J1)
      DO 210 I=1, NT3
      CP(I) = CPBB(J, I)
      CPY(I) = - CP(I)*SINB(I)
      CPZ(I) = - CP(I)*COSB(I)
  210 CONTINUE
      CALL SIMPSN( NT3, DTHETB, CP, CPS(J) )
      CALL SIMPSN( NT3, DTHETB, CPY, CPYS(J) )
      CALL SIMPSN( NT3, DTHETB, CPZ, CPZS(J) )
      CPS(J) = CPS(J)*RM*DRDX(J)
      CPYS(J) = CPYS(J)*RM
      CPZS(J) = CPZS(J)*RM
  200 CONTINUE

      CALL SIMPSN( NBD1, DX, CPS, CX)
      CALL SIMPSN( NBD1, DX, CPYS, CY)
      CALL SIMPSN( NBD1, DX, CPZS, CZ)
      CX = CX + 0.5*DX*( CPS(1) + CPS(NBD1) )
      CY = CY + 0.5*DX*( CPYS(1) + CPYS(NBD1) )
      CZ = CZ + 0.5*DX*( CPZS(1) + CPZS(NBD1) )
      IF ( YAW .GT. 0.01 ) GO TO 250
      CY = 0.0
      CX = 2.0*CX
      CZ = 2.0*CZ
  250 CONTINUE
C
      DO 300 J=1, NBD1
      CPZS(J) = CPZS(J) * XMIDB(J)
  300 CONTINUE
      CALL SIMPSN( NBD1, DX, CPZS, BBCM)
      BBCM = BBCM + 0.5*DX*( CPZS(1) + CPZS(NBD1) )
C
      BBCL = -CX*SINA*COSBT+ CY*(SING*COSA*COSG*SINBT*SINA) + CZ*
     1        (COSG*COSA -SING*SINBT*SINA)
      BBCD = CX*COSA*COSBT +  CY*(SING*SINA-COSG*SINBT*COSA) + CZ*
     1        (SING*SINBT*COSA + COSG*SINA)
      BBCD =  CX*SINBT + CY*COSG*COSBT - CZ*SING*COSBT
      BBCYM = BBCRM = 0.0
C
C
C
      RETURN
      END
      SUBROUTINE SIMPSN( NI, DELX, Y, SUM)
C
C
C          INTEGRATION BY SIMPSON RULE, EQUAL INTERVAL ONLY
C             N MUST BE .GE. 3
```

```
C
      DIMENSION Y(1)
C
      N = NI
      SUM = 0.0
      NS = 1
      NS1 = NS + 1
      IC = N - 2*(N/2)
      IF ( IC .EQ. 1 ) GO TO 90
      SUM = 3.0*DELX*( 3.0*( Y(2)+Y(3) ) + Y(1) + Y(4) )/8.0
      IF ( N .EQ. 4 ) GO TO 150
      NS = 4
      NS1 = NS + 1
   90 CONTINUE
C
      SUMEVN = 0.0
      SUMODD = 0.5*( Y(NS) - Y(N) )
      DO 100 I=NS1, N, 2
      SUMEVN = SUMEVN + Y(I)
      SUMODD = SUMODD + Y(I-1)
  100 CONTINUE
C
C
      SUM = SUM + 2.0*DELX*(SUMODD + 2.0*SUMEVN )/3.0
  150 CONTINUE
C
C
      RETURN
      END
      SUBROUTINE FORCEW(L,NB1,NC1,XC,XCR,YC,AREA,ARB,ALPHA,WCL,WCD,WCB,
     1                  WCPM,WCRM,WCYM)
C
C        COMPUTE THE FORCES AND MOMENTS ON THE WING
C
      COMMON/CONST /DR, RD, G, GP1, GM1, GR, AV1, AV2, AV3
      COMMON/WINGOT/ CPW,NWG,ISOLID,IFORM(10),YAW,BROLL,SLOPE(51)
      DIMENSION XC(1), XCR(1), YC(1), AREA(1),ALPHA(110,2),CPW(110,2)
C
      COSY = COS(YAW*DR)
      SINY = SIN(YAW*DR)
      COSA = COS(ARB*DR)
      SINA  = SIN(ARB*DR)
      TANA = TAN(ARB*DR)
      TANY = TAN(YAW*DR)
      A1 = 1.0/SQRT( 1.0 + TANA**2 + TANY**2 )
      COSB = A1*SQRT( 1.0 + TANA**2 )
      SINB = SQRT( 1.0 - COSB**2 )
      COSG = COS( BROLL*DR )
      SING = SIN( BROLL*DR )
      WCL = WCD = WCB = WCPM = WCRM = WCYM = 0.0
      NBC = NB1*NC1
C
      DO 100 I=1, NB1
      SUM = 0.0
      XLIFT = DRAG = BANK = 0.0
      DO 200 J=1, NC1
      IJ = J + (I-1)*NC1
      KJ = 1 + J +(I-1)*(NC1+1)
      SUM = SUM + AREA(IJ)
      COSAP = COS( ALPHA(KJ,L) )
      SINAP = SIN( ALPHA(KJ,L) )
      F = CPW(KJ, L) * AREA(IJ)
      FX = -F*SINAP
      FY = 0.0
      FZ = F*COSAP
      XL = -FX*SINA*COSB+ FY*(SING*COSA+COSG*SINB*SINA) + FZ*
     1        (COSG*COSA -SING*SINB*SINA)
      XD = FX*COSA*COSB +  FY*(SING*SINA-COSG*SINB*COSA) + FZ*
     1        (SING*SINB*COSA + COSG*SINA)
      XB =  FX*SINB + FY*COSG*COSB - FZ*SING*COSB
      XLIFT = XLIFT + XL
      DRAG = DRAG + XD
      BANK = BANK + XB
      WCPM = WCPM - XL*XCR(IJ)
      WCRM = WCRM + F*COSA*YC(IJ)
      WCYM = WCYM + XB*XCR(IJ) - XD*YC(IJ)
  200 CONTINUE
      WCL = WCL + XLIFT
      WCD = WCD + DRAG
      WCD = WCD + BANK
  100 CONTINUE
C
C
      RETURN
      END
      SUBROUTINE WRITEW(NC1, NB1, Z)
C           WRITE WING SURFACE SLOPE AND PRESSURE COEFFICIENT
C
      DIMENSION Z(1)
C
      NB2 = NB1 -1
      NC = NC1 + 1
      NBC2 = NB2-NC
      WRITE (6,801) ( I, I=1, NB1)
      WRITE (6,802)
      DO 100 J=1, NC1
      JS = J + 1
      JE = JS + NBC2
C
      WRITE (6, 803) ( J, (Z(I), I=JS,JE,NC)  )
  100 CONTINUE
C
  801 FORMAT (1H0,*   SPANWISE STATION*,I8,9I10/(T8X,I8,9I10))
  802 FORMAT (1H ,*   CHORDWISE STATION*)
  803 FORMAT (1H ,I10,F19.5,9F10.5/(F29.5,9F10.5) )
C
      RETURN
      END
      SUBROUTINE CUBERT
      DIMENSION X(3)
      COMMON/LOCTN / V(3), L, M, G(4), VMAX, VMIN
      COMMON/POINTS/ NPTS, NVAR
      L=0
      SUM = SQRT( G(1)**2+G(2)**2+G(3)**2+G(4)**2 )
      G1 = G(1) / SUM
      G2 = G(2) / SUM
      G3 = G(3) / SUM
      G4 = G(4) / SUM
C
C         EQ. 1   G1*X**3 + G2*X**2 + G3*X = 0.
      IF ( ABS( G4) .LT. 1.E-6) GO TO 4
C
C         EQ. 2   G1*X**3 + G2*X**2 + G3*X +G4 = 0.
      IF ( ABS( G1) .GT. 1.E-6) GO TO 9
C
C         EQ. 3   G2*X**2 + G3*X + G4 = 0.
      IF ( ABS( G2) .GT. 1.E-6) GO TO 7
C
C         EQ. 4   G3*X + G4 = 0.
      IF ( ABS( G3) .GT. 1.E-6) GO TO 5
      RETURN
C
    4 CONTINUE
      V(1) = 0.0
      L=1
      RETURN
    5 CONTINUE
      V(1) = - G(4) / G(3)
      GO TO 35
    7 .ONTINUE
      R1 = G(3)**2 - 4.0*G(2)*G(4)
      IF (  R1  .GT. 0.0 )  GO TO 107
      WRITE (6,901)
  901 FORMAT (1H0,*NAGATIVE RADICAL RESULTED FROM SOLVING THE QUADRATIC
     1EQUATION  G2.X2 + G3.X + G4 = 0*)
      RETURN
  107 CONTINUE
      R1=SQRT(R1)
      V1 = ( -G(3) + R1 ) *0.5 / G(2)
      V(1) = V1
      IF ( V1  .GT. VMIN .AND.  V1  .LT. VMAX ) GO TO 8
      V2 = (-G(3) - R1 )*0.5 / G(2)
      V(2) = V2
      IF ( V2  .GT. VMIN .AND.  V2  .LT. VMAX ) GO TO 8
      WRITE (6,902) VMIN, V1,V2, VMAX
  902 FORMAT (1H0,*BOTH ROOTS V1, V2 OF QUADRATIC EQ. G2.X2 + G3.X + G4
     1= 0 OUTSIDE OF LIMIT VMIN, VMAX*/ 1X,*VMIN =*,E14.4,4X,*V1 =*,
     2          E14.4,4X,*V2 =*,E14.4,4X,*VMAX =*,E14.4)
      RETURN
    8 CONTINUE
      L = 1
      RETURN
C
C        THREE ROOTS
C
    9 CONTINUE
      A1 = G(2) / G(1)
      A2=G(3)/G(1)
      A3=G(4)/G(1)
      A=A2-A1*A1/3.
      Q=2.*A1*A1*A1/27.-A1*A2/3.+A3
      RAD=Q*Q/4.+A*A*A/27.
      SIGN=1.
      IF(Q.GE.0.)SIGN=-1.
      IF(RAD)10,20,20
C
C                THREE UNEQUAL REAL ROOTS
C
   10 CONTINUE
      A3 = SQRT( ABS( A/3.0) )
      A2 = SQRT( ABS( A**3/3. ) )
      PHI = ACOS( ABS( 1.5*Q/ A2) )
      DO 11 I=1,3
      X(I) = 2.0*SIGN*A3*COS((PHI+6.2831853*FLOAT(I-1))/3.0)
   11 CONTINUE
      GO TO 40
C
C        EITHER ONE REAL AND TWO IMAG. OR THREE REAL ROOTS OF WHICH TWO
C        ARE EQUAL
C
   20 CONTINUE
      V1 = 1.0 / 3.0
      Q = A2*V1 - ( A1*V1)**2
      R = .5*A1*A2*V1 - .5 *A3 - ( A1*V1) **3
      RAD = Q**3 + R*R
      IF ( RAD  .GT.  0.0 )  GO TO 120
      WRITE (6,903)
  903 FORMAT (1HO,*ERROR IN TESTING RADICAL, REDUNDENT TEST*)
      RETURN
  120 CONTINUE
      RAD = SQRT( RAD)
      A2 = R + RAD
      SIGN = 1.
      IF ( A2  .LT.  0.0)  SIGN = -1.
      A2 = SIGN * ABS( A2 ) **V1
      A3 = R - RAD
      SIGN = 1.
      IF ( A3 .LT. 0.0 ) SIGN = -1.
      A3 = SIGN * ABS( A3 )**V1
      VO = A2 + A3 - A1*V1
   35 CONTINUE
      IF ( VO .GT. VMIN .AND. VO  .LT. VMAX ) L = 1
      IF ( L  .EQ. 0 ) WRITE (6,905)
  905 FORMAT (1HO,*THE ONLY UNEQUAL ROOT IS OUTSIDE OF VMIN, VMAX*/1X,
     1          *VMIN =*,E14.4,4X,* V =*,E14.4,4X,*VMAX =*,E14.4)
      V(1) = VO
      RETURN
   40 CONTINUE
      DO 45 I=1,3
      VO = X(I) - A1/3.0
      X(I) = VO
      V(I) = VO
      IF ( VO .LT. VMIN .OR.  VO  .GT. VMAX ) GO TO 50
      L=1
      RETURN
   50 CONTINUE
   45 CONTINUE
      WRITE (6,904) VMIN, V(1), V(2), V(3), VMAX
  904 FORMAT (1HO,*ALL THREE UNEQUAL ROOTS OUTSIDE OF LIMIT VMIN, VMAX*/
     1    1X,*VMIN =*,E14.4,4X,*V1 =*,E14.4,4X,*V2 =*,E14.4,4X,*V3 =*,
     2          E14.4,4X,*VMAX =*,E14.4)
C
      RETURN
      END
C
      SUBROUTINE POINT( M )
      COMMON/CURFIT/ B(2,4, 50), T( 50), SCALEF(2)
      COMMON/ ENDPT/ XFIRST, XLAST, XSAVE( 50)
      COMMON/LOCTN/ V(3), L, M, A(4), VMAX, VMIN
      COMMON/POINTS/ NPTS, NVAR
      COMMON/ POINTP/  X(4),  XP( 4),  XPP( 4)
      DIMENSION DX(4)
C
      N1 = NPTS - 1
      IF(N.NE.0)GO TO 4
      DO 1 I=1,N1
      M=I
      IF ( V(1)  .LE. T(I+1) ) GO TO 2
    1 CONTINUE
      WRITE(6,100)
  100 FORMAT(1HO,33H ERROR - PARAMETER EXCEEDS LIMITS)
      RETURN
    2 CONTINUE
      DO 300 K=1,3
      P = V(K)
      DX(K) = 1.E10
      IF ( P .LT. VMIN  .OR.  P  .GT. VMAX ) GO TO 300
      DX(K) = ABS((((B(1,1,M)*P + B(1,2,M))*P + B(1,3,M))*P + B(1,4,M))*
     1          SCALEF(1) - XIND )
  300 CONTINUE
```

```
      IF ( DX(1)  .GT. DX(2) ) GO TO 302
      IDENT = 1
      IF ( DX(1)  .GT. DX(3) )  IDENT = 3
      GO TO 305
  302 CONTINUE
      IDENT = 2
      IF ( DX(2)  .GT. DX(3) )  IDENT = 3
  305 CONTINUE
      P = V(IDENT)
C
      DO 3   I=1, NVAR
      X(I)=(((B(I,1,M)*P+B(I,2,M))*P+B(I,3,M))*P+B(I,4,M))*SCALEF(I)
      XP(I)=((3.*B(I,1,M)*P+2.*B(I,2,M))*P+B(I,3,M))*SCALEF(I)
    3 XPP(I)=(6.*B(I,1,M)*P+2.*B(I,2,M))*SCALEF(I)
C
  310 CONTINUE
      RETURN
C
    4 X(N)=X(N)/SCALEF(N)
      DO 5 I=1,N1
      M=I
      K = I + 1
      IF ( XSAVE(K)  .GE. X(N) ) GO TO 6
    5 CONTINUE
      XIND = X(N) * SCALEF(N)
      XFT = XFIRST * SCALEF(1)
      XLT = XLAST * SCALEF(1)
      WRITE (6,101)  XIND, XFT, XLT
  101 FORMAT (1H0,*  THE GIVEN VALUE OF INDEPENDENT VARIABLE *,E12.4,
     1           5X,*IS OUT OF RANGE TO BE INTERPOLATED  *,E12.4,5X,E12.4)
      RETURN
    6 DO 7 I=1,4
    7 A(I)=B(N,I,M)
      A(4)=A(4)-X(N)
      IF ( XFIRST  .EQ. X(N) ) GO TO 112
      IF ( XLAST   .EQ. X(1) ) GO TO 111
      VMAX = T(M+1) * 1.0001
      VMIN = T(M) * 0.9999
      CALL CUBERT
      XIND = X(1)* SCALEF(1)
      GO TO 103
  112 V(1) = 0.0
      L=1
      GO TO 103
  111 V(1) = 1.0
      L=1
  103 CONTINUE
      IF(L.NE.0)GO TO 2
      OP = P
      WRITE (6,10) OP
   10 FORMAT (1X4HOP =,E15.8)
      DO 8 J=1,2
      K = M + J - 1
      B1 = B(N,1,K)
      B2 = B(N,2,K)
      B3 = B(N,3,K)
      B4 = B(N,4,K)
      P = T(K)
      SUM - B1*P**3 + B2*P**2 + B3*P + B4
      DIF = SUM - X(N)
    8 WRITE (6,9) K,P,B1,B2,B3,B4,X(N),SUM,DIF
    9 FORMAT (1XI3,8E15.6)
      WRITE(6,102)
  102 FORMAT(1H0,32H FAILURE IN CUBE ROOT EXTRACTION)
      RETURN
      END
C
      SUBROUTINE SPLFIT
      REAL L,M,MU
      COMMON/CURFIT/ B(2,4, 50), T( 50), SCALEF(2)
      COMMON/ ENDPT/ XFIRST, XLAST, XSAVE( 50)
      COMMON/INPUT / X( 2,  50)
      COMMON/POINTS/ NPTS, NVAR
      DIMENSION M(2, 50), S( 50), L( 50), MU( 50), D( 50), P( 50), Q( 50
     1         ), U( 50)
      IF(NPTS .GT. 50) NPTS= 50
      DO 200 I=1, NVAR
  200 SCALEF(I)=0.
      DO 201 I=1,NPTS
      DO 201 J=1,NVAR
  201 SCALEF(J)=AMAX1(SCALEF(J),ABS(X(J,I)))
      DO 101 J=1,NVAR
      IF ( SCALEF(J)  .LT. 1.E-10 ) SCALEF(J) = 1.0
  101 CONTINUE
      DO 202 I=1,NPTS
      DO 202 J=1,NVAR
  202 X(J,I)=X(J,I)/SCALEF(J)
      DO 203 I=1,NPTS
      XSAVE(I) = X(1,I)
  203  CONTINUE
      XFIRST = X(1,1)
      XLAST = X(1,NPTS)
      T(1)=0.
      S(1)=0.
      SUM=0.
      L(1)=-2.
      MU(1)=0.
      L(NPTS)=0.
      MU(NPTS)=-2.
      D(1)=0.
      D(NPTS)=0.
      P(1)=2.
      Q(1)=1.
      U(1)=0.
      N1=NPTS-1
      DO 2 I=2,NPTS
      SUM1=0.
      DO 1 J=1,NVAR
    1 SUM1=SUM1+(X(J,I)-X(J,I-1))**2
      S(I)=SQRT(SUM1)
    2 T(I)=T(I-1)+S(I)
      DO 3 I=2,NPTS
      S(I)=S(I)/T(NPTS)
    3 T(I)=T(I)/T(NPTS)
      DO 4 I=2,N1
      L(I)=S(I+1)/(S(I)+S(I+1))
      MU(I)=1.-L(I)
      P(I)=MU(I)*Q(I-1)+2.
    4 Q(I)=-L(I)/P(I)
      P(NPTS)=2.*(1.-Q(N1))
      DO 7 I=1,NVAR
      DO 5 J=2,N1
      D(J)=6.*((X(I,J+1)-X(I,J))/S(J+1)-(X(I,J)-X(I,J-1))/S(J))/(S(J)+
     1S(J+1))
    5 U(J)=(D(J)-MU(J)*U(J-1))/P(J)
      U(NPTS)=-MU(NPTS)*U(N1)/P(NPTS)
      M(I,NPTS)=U(NPTS)
      DO 6 J=1,N1
      K=NPTS-J
      M(I,K)=Q(K)*M(I,K +1)+U(K)
  B(I,1,K)=(M(I,K +1)-M(I,K))/(6.*S(K+1))
  B(I,2,K)=.5*(M(I,K)*T(K+1)-M(I,K+1)*T(K))/S(K+1)
  B(I,3,K)=(.5*(P(I,K+1)*T(K)**2-M(I,K)*T(K+1)**2)+X(I,K+1)-X(I,K)+
 1(M(I,K)-M(I,K+1))*S(K+1)**2/6.)/S(K+1)
6 B(I,4,K)=(M(I,K)*T(K+1)**3-M(I,K+1)*T(K)**3)/(6.*S(K+1))+(X(I,K)*T
 1(K+1)-X(I,K+1)*T(K))/S(K+1)+S(K+1)/6.*(M(I,K+1)*T(K)-M(I,K)*T(K+1)
 1)
7 CONTINUE
  RETURN
  END
```

DISTRIBUTION LIST

	Copies
Navy	
Commander	
Naval Surface Weapons Center	
Dahlgren, Virginia 22448	
G05 (Soper)	1
G40	1
G402	30
G41	1
G402 (Matra)	32
G42 (Graff)	1
N11 (Hardy)	1
K21 (Blackman)	1
K82 (Edwards)	1
X211	3
S20 (Reinhard)	1
K21 (Purvis)	1
R44 (Baltakis)	1
K21 (Moore)	1
Office of Chief of Naval Operations	
Operations Evaluation Group (OP-03EG)	
Washington, D. C. 20350	1
Defense Nuclear Agency	
Washington, D. C. 20301	2
Commander	
Naval Sea Systems Command	
Washington, D. C. 20362	
SEA-03A	
SEA-0333	1
SEA-035-13	2
SEA-035A	1
SEA-55212	1
SEA-0632	2
SEA-654312	1
PMS-403-02	1
Commander	
Naval Air Systems Command (AIR-320-B)	
Washington, D. C. 20361	1
Commander	
Pacific Missile Test Center	
Point Mugu, California 93042	
T. W. Elliott (Code 5331)	1

PRECEDING PAGE NOT FILMED
BLANK

	Copies
Commander	
Naval Air Development Center	
Johnsville, Warminister, Pennsylvania 18974	
Dr. E. McQuillan (Code 3033)	1
Herman Rubin (Code 3033)	1
K. T. Yen (Code 30331)	1
Naval Research Laboratory	
Washington, D. C. 20390	
Dr. M. R. Achter (Code 6340)	1
Commander	
Naval Weapons Center	
China Lake, California 93557	
Code 4062	1
Code 45703	1
Mr. E. L. Jeter	1
James E. Serpanos	1
Office of Naval Research	
Department of the Navy	
North Quincy Street	
Arlington, Virginia 22217	
Code 461	1
Code 439 (N. Perrone)	1
Code 470	1
Technical Library	2
Naval Postgraduate School	
Monterey, California 93940	
Code 2124	1
Dr. J. G. Cantin	1
Commanding Officer	
Naval Ship Research and Development Center	
Bethesda, Maryland 20034	
Code 1844 (Jim McKee)	1
Code 1844 (Gordon Everstine)	1
Code 564	2
Air Force	
AFFDL/PTS (CAPT Gilbert L. Camburn)	
AF Flight Dynamics Laboratory	
Wright-Patterson Air Force Base, Ohio 45433	1
AF Materials Laboratory	
Wright-Patterson Air Force Base, Ohio 45433	
Code MAAN (B. R. Emrich)	1

Copies

Flight Dynamics Laboratory
Structures Division
Wright-Patterson Air Force Base, Ohio 45433
Mr. Leo J. D. Bernier 1

AFML/LLM (W. G. Ramke)
Wright-Patterson Air Force Base, Ohio 45433 1

AFATL/DLRV
Eglin Air Force Base, Florida 32542
James M. Heard 2

AFAL/TEM-3
Wright-Patterson Air Force Base, Ohio 45433
Richard A. Ireland 1

Army

Commanding General
U. S. Army Missile Command
Redstone Arsenal, Alabama 35809 1

Other Agencies

NASA Scientific and Technical Information Facility
P.O. Box 33
College Park, Maryland 20740
Acquisitions Branch - Foreign Exchange 2

Industry

Aerospace Corporation
1111 East Mill Street
San Bernardino, California 92408
Dr. Robert L. Hallse 1

Booz Allen Applied Research, Inc.
P.O. Box 874
Shalimar, Florida 32579
William Day 1

Applied Physics Laboratory
The Johns Hopkins University
Johns Hopkins Road
Laurel, Maryland 20810
L. B. Weckesser 1
W. C. Caywood 1
R. M. Rivello 1

	Copies
Battelle Memorial Institute	
Defense Metals Information Center	
505 King Avenue	
Columbus, Ohio 43201	
H. D. Moran	1
Brunswick Corporation	
Technical Products Division	
325 Brunswick Lane	
Marion, Virginia 24354	
Robert Copeland	1
General Dynamics/Pomona	
P.O. Box 2507	
Pomona, California 91766	
R. A. Miller	1
D. A. Underhill	1
University of Maryland	
College Park, Maryland 20740	
Dr. Schaeffer	1
Raytheon Company	
Missile Systems Division, VB1-3	
Hartwell Road	
Bedford, Massachusetts 01730	
R. D. Howe	1
D. H. Sanders	1
McDonnell Douglas Astronautics Co.	
5301 Bolsa Avenue	
Huntington Beach, California 92647	
W. F. Bozich	1
K. K. Wang	1
R. S. Lee	1
McDonnell Douglas Astronautics Co.	
Eastern Division	
P.O. Box 516	
St. Louis, Missouri 63166	
J. Gubser	1
Lockheed Missiles and Space Co., Inc.	
Dept S1-12, Building 154	
Sunnyvale, California 94088	
J. A. Bailie	1

	Copies
Lockheed Missile and Space Co., Inc.	
Dept 52-30, Building 103	
P.O. Box 504	
Sunnyvale, California 94088	
Robert M. Beasley	1
Senior Staff Scientist	1
Lockheed Missile and Space Co., Inc.	
Technical Information Center	
3251 Hanover Street	
Palo Alto, California 94034	1
Hughes Aircraft Company	
Missile System Division	
8433 Fallbrook Avenue	
Canoga Park, California 91304	
R. J. Oedy	1
Hughes Aircraft Company	
Centinela and Teale Streets	
Culver City, California 90230	
L. E. Gates, M.S. H-164	1
The Boeing Company	
P.O. Box 3999	
Seattle, Washington 98124	
R. Blaisdell	1
K. Brettmann	1
Martin-Marietta Corporation	
P. O. Box 5837	
Orlando, Florida 32855	
G. H. Sprague, Jr. (Director Aeormechanical Division)	1
G. Fotieo	1
Honeywell, Incorporated	
600 2nd Street North	
Hopkins, Minnesota 55343	
J. D. Brannan	1
Fairchild Republic Co.	
Farmingdale, New York 11735	
Jim Clarke	1
Mail Stop Z-19A-08	1
Raytheon Co.	
Spenser Lab	
Burlington, Massachusetts 01803	
Doris A. Holden	1
Dept 7113	1
TOTAL	150